This document is based on European standards and is not valid for use in U.S.A.

Compact / CANopen / Logic Controller / Twido
System User Guide

EIO0000000275

MAR 2010

Schneider Electric

Contents

Important Information 3
Before You Begin 4
Introduction 6
Glossary 8
Application Source Code 9
Typical Applications 10
System 11
Architecture 11
Installation 14
Hardware 17
Software 24
Communication 25
Implementation 29
Communication 30
Controller 33
HMI 52
Devices 65
Altivar 312 65
Appendix 77
Detailed Component List 77
Component Protection Classes 79
Component Features 80
Contact 85

Important Information

NOTICE Read these instructions carefully, and look at the equipment to become familiar with the device before trying to install, operate, or maintain it. The following special messages may appear throughout this documentation or on the equipment to warn of potential hazards or to call attention to information that clarifies or simplifies a procedure.

The addition of this symbol to a Danger or Warning safety label indicates that an electrical hazard exists, which will result in personal injury if the instructions are not followed.

This is the safety alert symbol. It is used to alert you to potential personal injury hazards. Obey all safety messages that follow this symbol to avoid possible injury or death.

⚠ DANGER

DANGER indicates an imminently hazardous situation, which, if not avoided, **will result in** death or serious injury.

⚠ WARNING

WARNING indicates a potentially hazardous situation, which, if not avoided, **can result in** death, serious injury, or equipment damage.

⚠ CAUTION

CAUTION indicates a potentially hazardous situation, which, if not avoided, **can result in** injury or equipment damage.

PLEASE NOTE Electrical equipment should be installed, operated, serviced, and maintained only by qualified personnel. No responsibility is assumed by Schneider Electric for any consequences arising out of the use of this material.

A qualified person is one who has skills and knowledge related to the construction and operation of electrical equipment and the installation, and has received safety training to recognize and avoid the hazards involved

© 2009 Schneider Electric. All Rights Reserved.

Before You Begin

Do not use this product on machinery lacking effective point-of-operation guarding. Lack of effective point-of-operation guarding on a machine can result in serious injury to the operator of that machine.

⚠ WARNING
UNGUARDED MACHINERY CAN CAUSE SERIOUS INJURY • Do not use this software and related automation products on equipment which does not have point-of-operation protection. • Do not reach into machine during operation. **Failure to follow these instructions can cause death, serious injury or equipment damage.**

This automation equipment and related software is used to control a variety of industrial processes. The type or model of automation equipment suitable for each application will vary depending on factors such as the control function required, degree of protection required, production methods, unusual conditions, government regulations, etc. In some applications, more than one processor may be required, as when backup redundancy is needed.

Only the user can be aware of all the conditions and factors present during setup, operation and maintenance of the machine; therefore, only the user can determine the automation equipment and the related safeties and interlocks which can be properly used. When selecting automation and control equipment and related software for a particular application, the user should refer to the applicable local and national standards and regulations. A “National Safety Council’s” Accident Prevention Manual also provides much useful information.

In some applications, such as packaging machinery, additional operator protection such as point-of-operation guarding must be provided. This is necessary if the operator’s hands and other parts of the body are free to enter the pinch points or other hazardous areas and serious injury can occur. Software products by itself cannot protect an operator from injury. For this reason the software cannot be substituted for or take the place of point-of-operation protection.

Ensure that appropriate safeties and mechanical/electrical interlocks for point-of-operation protection have been installed and are operational before placing the equipment into service. All mechanical/electrical interlocks and safeties for point-of-operation protection must be coordinated with the related automation equipment and software programming.

NOTE: Coordination of safeties and mechanical/electrical interlocks for point-of-operation protection is outside the scope of this document.

START UP AND TEST

Before using electrical control and automation equipment for regular operation after installation, the system should be given a start up test by qualified personnel to verify correct operation of the equipment. It is important that arrangements for such a check be made and that enough time is allowed to perform complete and satisfactory testing.

⚠ CAUTION
EQUIPMENT OPERATION HAZARD • Verify that all installation and set up procedures have been completed. • Before operational tests are performed, remove all blocks or other temporary holding means used for shipment from all component devices. • Remove tools, meters and debris from equipment. **Failure to follow these instructions can result in injury or equipment damage.**

Follow all start up tests recommended in the equipment documentation. Store all equipment documentation for future reference.

Software testing must be done in both simulated and real environments.

Verify that the completed system is free from all short circuits and grounds, except those grounds installed according to local regulations (according to the National Electrical Code in the U.S.A, for instance). If high-potential voltage testing is necessary, follow recommendations in equipment documentation to prevent accidental equipment damage.

Before energizing equipment:

- Remove tools, meters, and debris from equipment.
- Close the equipment enclosure door.
- Remove ground from incoming power lines.
- Perform all start-up tests recommended by the manufacturer.

OPERATION AND ADJUSTMENTS

The following precautions are from NEMA Standards Publication ICS 7.1-1995 (English version prevails):

- Regardless of the care exercised in the design and manufacture of equipment or in the selection and rating of components, there are hazards that can be encountered if such equipment is improperly operated.

- It is sometimes possible to misadjust the equipment and thus produce unsatisfactory or unsafe operation. Always use the manufacturer's instructions as a guide for functional adjustments. Personnel who have access to these adjustments should be familiar with the equipment manufacturer's instructions and the machinery used with the electrical equipment.

- Only those operational adjustments actually required by the operator should be accessible to the operator. Access to other controls should be restricted to prevent unauthorized changes in operating characteristics.

⚠ WARNING
UNEXPECTED EQUIPMENT OPERATION • Only use software tools approved by Schneider Electric for use with this equipment. • Update your application program every time you change the physical hardware configuration. **Failure to follow these instructions can cause death, serious injury or equipment damage.**

Introduction

Introduction

This document is intended to provide a quick introduction to the described system. It is not intended to replace any specific product documentation, nor any of your own design documentation. On the contrary, it offers additional information to the product documentation, for installing, configuring and implementing the system.

The architecture described in this document is not a specific product in the normal commercial sense. It describes an example of how Schneider-Electric and third-party components may be integrated to fulfill an industrial application.

A detailed functional description or the specification for a specific user application is not part of this document. Nevertheless, the document outlines some typical applications where the system might be implemented.

The architecture described in this document has been fully tested in our laboratories using all the specific references you will find in the component list near the end of this document. Of course, your specific application requirements may be different and will require additional and/or different components. In this case, you will have to adapt the information provided in this document to your particular needs. To do so, you will need to consult the specific product documentation of the components that you are substituting in this architecture. Pay particular attention in conforming to any safety information, different electrical requirements and normative standards that would apply to your adaptation.

It should be noted that there are some major components in the architecture described in this document that cannot be substituted without completely invalidating the architecture, descriptions, instructions, wiring diagrams and compatibility between the various software and hardware components specified herein. You must be aware of the consequences of component substitution in the architecture described in this document as substitutions may impair the compatibility and interoperability of software and hardware.

⚠ CAUTION

EQUIPMENT INCOMPATIBILITY OR INOPERABLE EQUIPMENT

Read and thoroughly understand all hardware and software documentation before attempting any component substitutions.

Failure to follow these instructions can result in injury or equipment damage.

Abbreviations

Abbreviation	Signification
AC	Alternating Current
CB	Circuit Breaker
DI	Digital Input
DO	Digital Output
DC	Direct Current
EDS	Electronic Data Sheet
E-STOP	Emergency Stop
HMI	Human Machine Interface
I/O	Input/Output
IL	Instruction List – a textual IEC-61131 programming language
LD	Ladder Diagram – a graphic IEC-61131 programming language
PC	Personal Computer
PDO	Process Data Object (CANopen)
PS	Power Supply
RPM	Revolutions Per Minute
RPDO	Receive Process Data Object (CANopen)
SE	Schneider Electric
SDO	Service Data Object
TPDO	Transmit Process Data Object (CANopen)
TVDA	Tested, Validated, Documented Architectures
VSD	Variable Speed Drive
WxHxD	Dimensions : Width, Height and Depth

Glossary

Expression	Signification
Altivar (ATV)	SE product name for a family of VSDs
CANopen	Name for a communications machine bus system
Harmony	SE product name for a family of switches and indicators
Magelis	SE product name for a family of HMI-Devices
MB - SL	SE name for a serial Modbus communications protocol
Modbus	A Communications protocol
OsiSense	SE product name for a family of sensors
Phaseo	SE product name for a family of power supplies
SoMove	An SE software product for configuring drives
Preventa	SE product name for a family of safety devices
TeSys	SE product name for a family for motor protection devices and load contactors
Twido	SE product name of a basic range family of Controllers
TwidoSuite	SE product name for a Controller programming software
Vijeo Designer Lite	An SE software product for programming Magelis HMI devices

Application Source Code

Introduction

Examples of the source code and wiring diagrams used throughout this document can be downloaded from our website.

The example source code is in the form of configuration, application and import files. Use the appropriate software tool to either open or import the files.

Extension	File Type	Software Tool Required
CFG	Export File (Multiloader Export)	SoMove (Lite)
DOC	Document file	Microsoft Word
DOP	Project File	Vijeo Designer Lite
EDS	**E**lectronic **D**ata **S**heet – Device Definition	Industrial standard
PDF	Portable Document Format - document	Adobe Acrobat
PSX	Project file	SoMove (Lite)
SPA	Schneider Product Archive	TwidoSuite
XPR	Project file	TwidoSuite
Z13	Project file	EPLAN

Typical Applications

Introduction

Here you will find a list of the typical applications, and their market segments, where this system or subsystem can be applied:

Packaging:

- Filling machines
- Bottling machines
- Wrapping machines
- Corking machines
- Pallet wrappers

Textile:

- Clothing machines
- Sawing machines

Pumping:

- Booster stations
- Compressors
- Vacuum pumps
- Hydraulic, Air, Filter pumps…

HVAC-R:

- Compressors

Other Machines:

- Wood working machines
- Cutting machines
- Sanders
- Sawing machines
- Industrial washing machines

System

Introduction

The system chapter describes the architecture, dimensions, quantities, required software and the different types of components used within this system.

Architecture

General

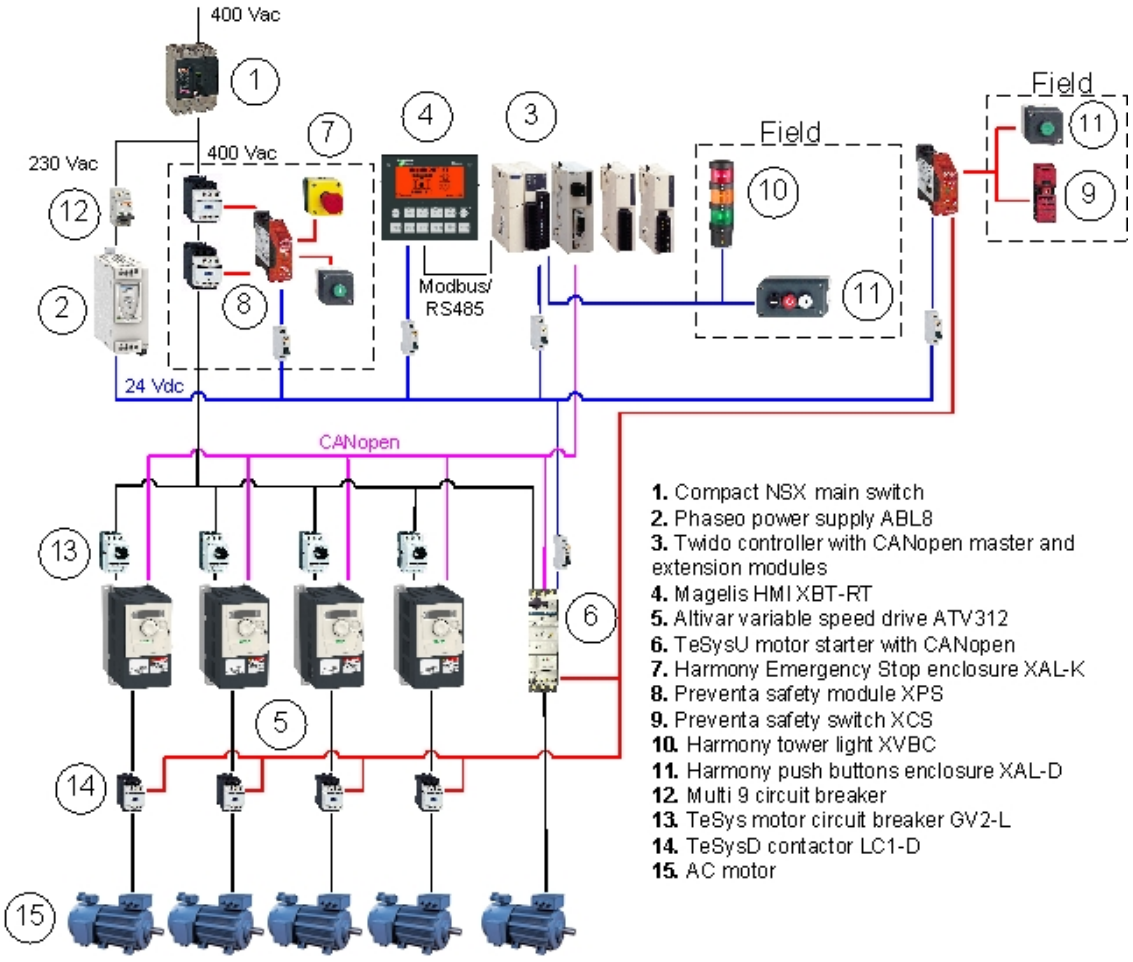
The controller in this application is a Twido. The user can control the application using the Magelis HMI.

The variable speed drives, connected to the Twido via CANopen bus, are of the type Altivar 312. An additional motor is driven by a TeSysU motor starter.

The example application includes two functional safety options:

1. A tamper free Emergency Stop function supervised with a Preventa safety module.
2. A second safety module to evaluate protective door guard function.

Layout



Components

Hardware:

- Compact main switch NSX100F
- Motor circuit breaker GV2L
- TeSysD contactor LC1D
- Altivar variable speed drive ATV312
- TeSysU motor starter
- Harmony XALK Emergency Stop switch with rotation release
- Phaseo power supply ABL8
- Modicon Twido modular controller
- CANopen tap TSX
- Magelis XBTRT HMI touch operator terminal
- Harmony XB5 selector switches, push buttons and indicator lamps
- Harmony tower light
- Preventa guard switch with mechanical actuator XCS
- Preventa safety module XPS
- Multi 9 circuit breaker
- Standard AC motor

Software:

- Twidosuite Version 2.20.10
- SoMove 1.1.6.0
- Vijeo-Designer Lite 1.2

Quantities of Components

For a complete and detailed list of components, the quantities required and the order numbers, please refer to the detailed components list in the appendix.

Degree of Protection

Not all the components in this configuration are designed to withstand the same environmental conditions. Some components may need additional protection, in the form of housings, depending on the environment in which you intend to use them. For environmental details of the individual components please refer to the list in the appendix of this document and the appropriate user manual.

Cabinet Technical Data

Input	Mains voltage	400 Vac
	Power requirement	~ 3 kW
	Cable Size	5 x 2.5 mm² (L1, L2, L3, N, PE)
	Cable Connection	3 phase + Neutral + Ground Neutral is needed for 230 Vac (Phase and Neutral)
Output	Motor power ratings	4 asynchronous motors (4 poles:1500 RPM) controlled by ATV312 (0.37kW) 1 asynchronous motors (4 poles:1500 RPM) controlled by TeSysU (0.18kW)

Functional Safety Notice

(EN ISO13849-1 EN IEC62061)

The standard and level of functional safety you apply to your application is determined by your system design and the overall extent to which your system may be a hazard to people and machinery.

As there are no moving mechanical parts in this application example, category 1 (according to EN13849-1) has been selected as an optional safety level.

Whether or not this functional safety category should be applied to your system should be ascertained with a proper risk analysis.

This document is not comprehensive for any systems using the given architecture and does not absolve users of their duty to uphold the functional safety requirements with respect to the equipment used in their systems or of compliance with either national or international safety laws and regulations

Emergency Stop

Emergency Stop/Emergency Disconnection function

This function for stopping in an emergency is a protective measure which complements the safety functions for the safeguarding of hazardous zones according to prEN ISO 12100-2.

Safety Function

Door guarding :

up to Performance Level (PL) = b , Category 1, Safety Integrity Level (SIL) = 1

Dimensions

The compact dimensions of the devices used, for example, the controller and power supply, enable the components to be installed inside a small control panel with the following external dimensions: 1200 x 800 x 400 mm (WxHxD).

The display elements used to indicate “Plant Running” and “Safety Acknowledged” can be built into the door of the control cabinet along with the system master switch and Emergency Stop switch.

Installation

Introduction

This chapter describes the steps necessary to set up the hardware and configure the software required to fulfill the described function of the application.

Assembly

Notes

The components and I/O points listed below represent a cross-section of the components and signals which can be used in conjunction with most typical applications (functional safety/maintenance switches) and are a basic necessity for control and display purposes and a number of optional inputs and outputs, .

- The components designed for installation in a control cabinet, example Twido , Phaseo power supply unit, Harmony Emergency Stop switching device, line circuit breaker, contactors and motor circuit breaker, etc. can be snapped onto a 35 mm DIN rail.
- The Altivar variable speed drive is installed directly on the mounting plate.
- Emergency Stop, main switch and HMI are installed in the cabinet door.
- The Preventa guard switch is mounted in the field.
- There are two options available for mounting Harmony XB5 push buttons and indicator lamps:

    1. Using a 22 mm hole drilled into the front door of the control cabinet in the appropriate position.
    2. Using an XALD housing, which can house up to 5 push buttons or indicator lamps. This XALD is designed for backplane assembly or direct wall mounting.

- 400 Vac wiring between Compact main switch, motor circuit breaker, load relay and variable speed drives.
- 24 Vdc wiring between power supply unit, Twido, push buttons, indicator lamps and variable speed drive control circuit.
- 24 Vdc between Emergency Stop, Preventa, and the control circuit of the load contactor.

Twido I/O Wiring

Twido inputs	Description
I 0.0	Not used, reserved for fast counters
I 0.1	Not used, reserved for fast counters
I 0.2	Safety power
I 0.3	Safety drives
I 0.4	Motor protection switch drive 1
I 0.5	Motor protection switch drive 2
I 0.6	Motor protection switch drive 3
I 0.7	Motor protection switch drive 4
I 0.8	Button start
I 0.9	Button stop
I 0.10	--Reserved--
I 0.11	--Reserved--

Twido outputs	Description
Trans. Out Q0	Safety power actuated
Trans. Out Q1	Safety drives actuated
Relay Out Q2	start
Relay Out Q3	stop
Relay Out Q4	plant ready for operation
Relay Out Q5	--Reserved--
Relay Out Q6	--Reserved--
Relay Out Q7	--Reserved--

Digital extension module

Digital I/O-Module inputs	Description
all inputs	--Reserved--

Digital I/O-Module outputs	Description
Relay Out Q0	green indicator lamp
Relay Out Q1	red indicator lamp
Relay Out Q2	blue lamp
Relay Out Q3	white lamp
Relay Out Q4	--Reserved--
Relay Out Q5	--Reserved--
Relay Out Q6	--Reserved--
Relay Out Q7	--Reserved--

Analog extension module

Analog I/Os	Description
all inputs and outputs	--Reserved--

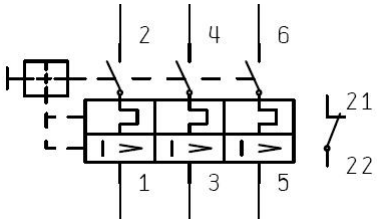
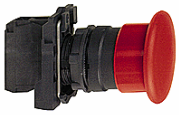
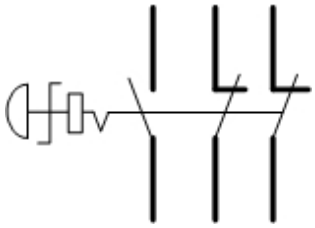
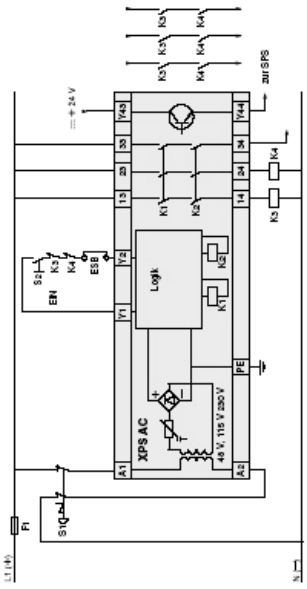
Thermo-couple module

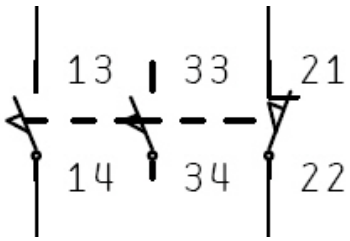
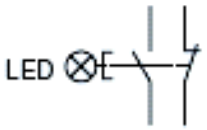
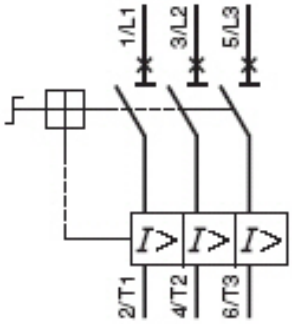
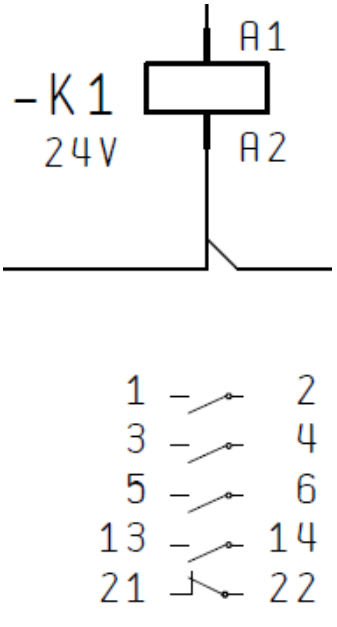
Connections	Description
Out0	--Reserved--
In0	Temperature sensor Pt100
In1	Temperature sensor Pt100

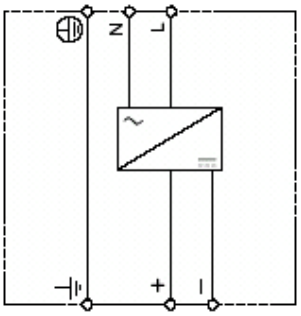
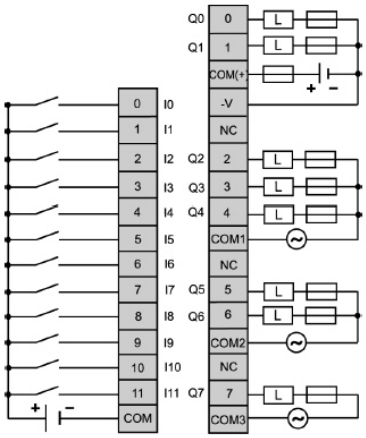
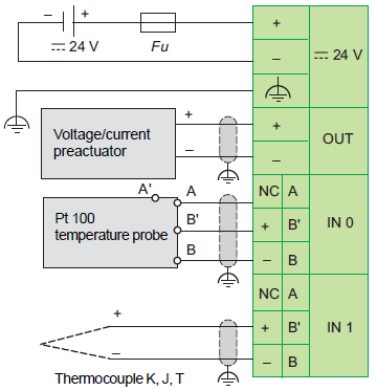
Twido CANopen module	Description
Com	CANopen bus communication

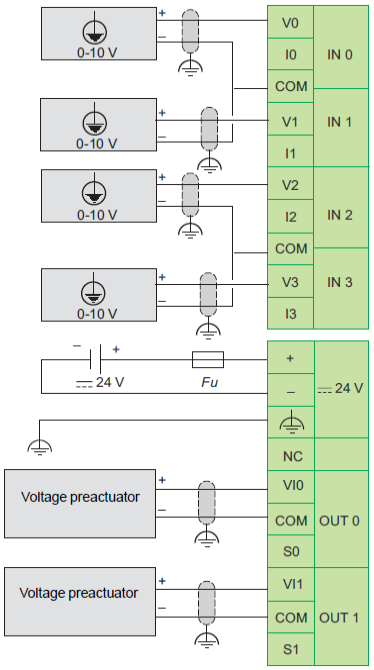
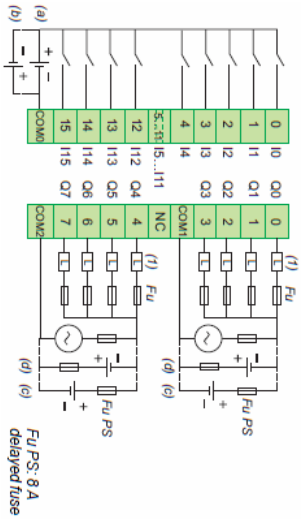
Twido 24 Vdc supply	Description
COM	Supply inputs 0 Vdc
COM+	Supply outputs +24 Vdc
-V	Supply outputs 0 Vdc
COM 1	Supply outputs +24 Vdc
COM 2	Supply outputs +24 Vdc
COM 3	Supply outputs +24 Vdc

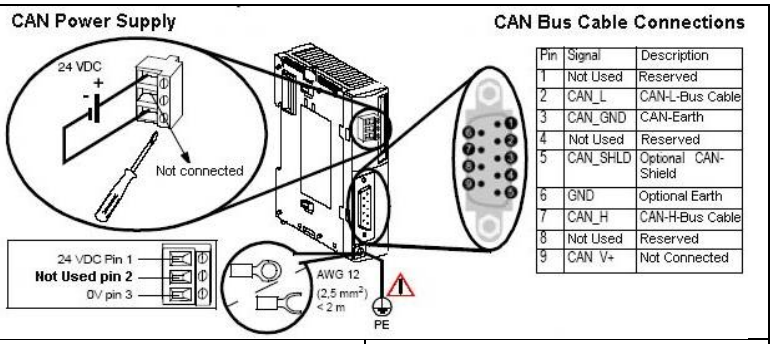
Hardware

Main Switch **Compact** **NSX100F**		
Emergency Stop for door mounting **Harmony** **XB5AS844 + XB5AZ141** Incl. Emergency Stop Label **ZBY8330**	 	
Emergency Stop Safety Module **Preventa** **XPSAC5121**		

Guard switch **Preventa** **XCSA502**		
Indicator Press Button **Harmony** **XB5**		
Motor circuit breaker **TeSys** **GV2L07**		
Contactor **TeSysD** **LC1D09BD**		

Power supply **Phaseo** **ABL8RPS24050** 24 Vdc – 5 A		
Modicon Twido **Logic Controller** Modular Device **TWDLMDA20DRT**		
J,K,T thermocouple **12 bit** 2x In + 1x Out **Twido** **TM2ALM3LT**		

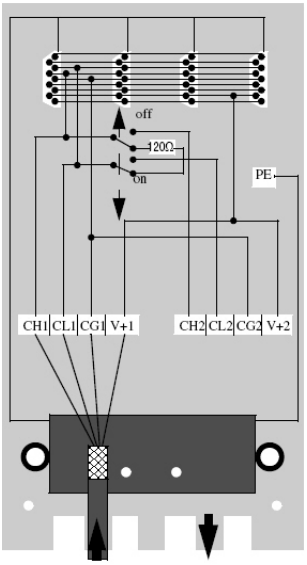
Analog module 4x In + 2x Out **Twido** **TM2AMM6HT**		
Digital module 16x In + 8x Out **Twido** **TM2DMM24DRF**		
Interface Module **CANopen Master** **Twido** **TWDNCO1M**		

<table>
<tr>
<td></td>
<td colspan="2"> 

Pin	Signal	Description
1	Not Used	Reserved
2	CAN_L	CAN-L-Bus Cable
3	CAN_GND	CAN-Earth
4	Not Used	Reserved
5	CAN_SHLD	Optional CAN-Shield
6	GND	Optional Earth
7	CAN_H	CAN-H-Bus Cable
8	Not Used	Reserved
9	CAN V+	Not Connected

</td>
</tr>
<tr>
<td>CANopen-TAP

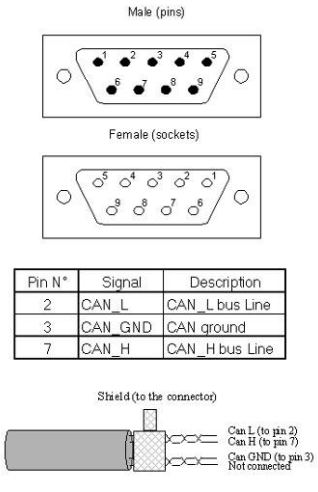
TSXCANTDM4

(4-port adapter)</td>
<td>
</td>
<td> 
</td>
</tr>
<tr>
<td>CANopen – connector

TSXCANKCFD180T

with integrated terminal resistor for linking to the Twido-CANopen-Master

CANopen – Cable

TSXCANCA50</td>
<td> </td>
<td> 

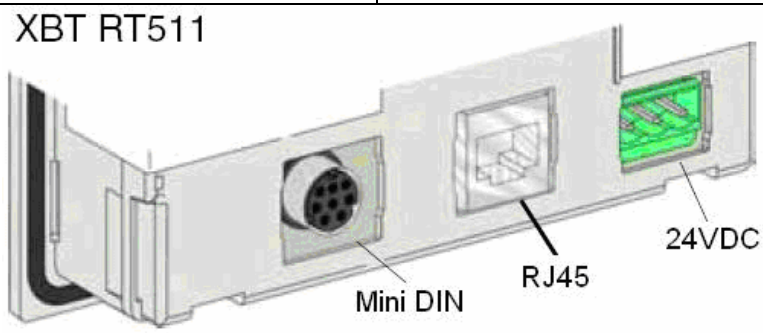
Pin N°	Signal	Description
2	CAN_L	CAN_L bus Line
3	CAN_GND	CAN ground
7	CAN_H	CAN_H bus Line

</td>
</tr>
</table>

Magelis

HMI Touch-Display and front-keys

XBTRT511

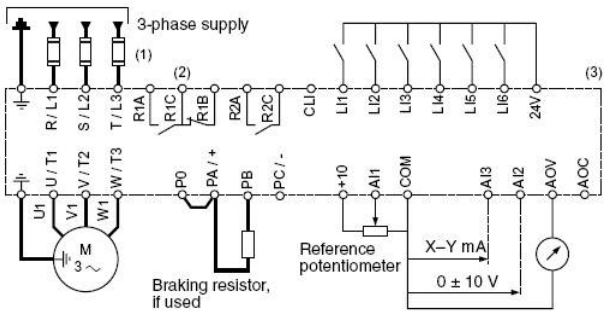


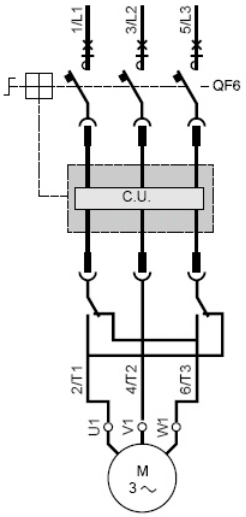
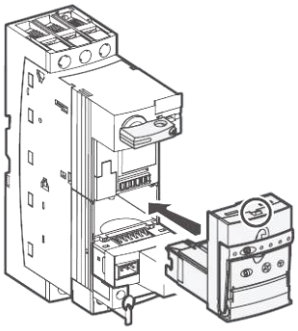
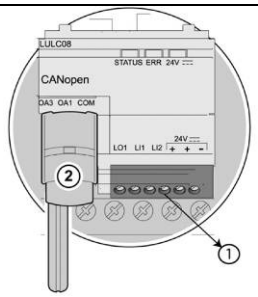
Variable Speed Drive

Altivar 312

ATV312H037N4

3-phase,
400 Vac – 0.37 kW



Motor Starter **TeSysU** Power base two directions **LUB12BL** Coil wiring kit **LULC08**		
Motor Starter **TeSysU** Control Unit **LUCA05BL**		
Motor Starter **TeSysU** CANopen communication module **LULC08**		 1. 24 Vdc power Supply 2. Terminal for coil wiring kit

Software

General

TwidoSuite is used for programming the Twido, including the configuration for communication and assigning inputs and outputs.

The HMI application on the Magelis operator terminal is configured using Vijeo-Designer Lite software.

The Altivar 312 variable speed drive can be configured using the keypad panel. However, the SoMove software is a more user-friendly option and can be used for parameterizing the drive, saving data and quickly restoring existing data for service purposes. The software can also help you to optimize the parameters online.

To use the software packages, your PC must have the appropriate Microsoft Windows operating system installed:

- Windows XP Professional

The default installation path on the hard drive of your PC for each of the software tools is:

TwidoSuite

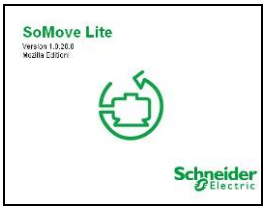
C:\Program Files\Schneider Electric\TwidoSuite

Vijeo-Designer Lite

C:\Program Files\Schneider Electric\Vijeo-Designer Lite

SoMove Lite

C:\Program Files\Schneider Electric\SoMove Lite



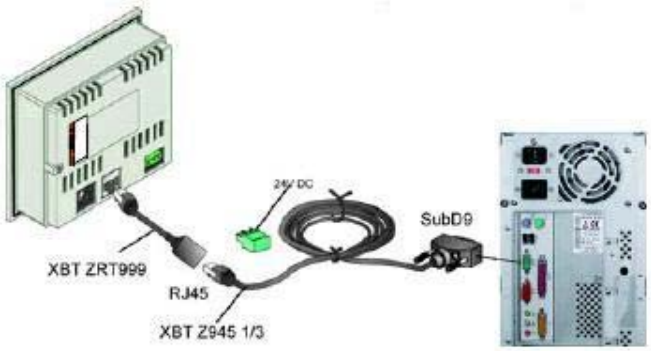
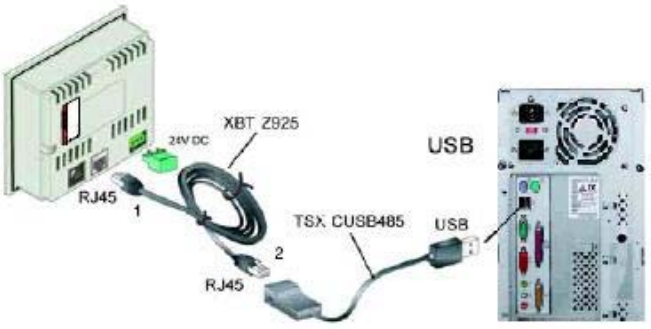
Communication

General

A Modbus connection is used to exchange data between the Magelis HMI terminal and the Twido. The XBTZ9780 communication cable shown below is required to connect these two devices. The software driver required for Modbus communication is already included in the software packages for the Magelis panel and the Twido.

This chapter describes the hardware used for data communications (CANopen) and how to use the programming cables used in configuring the individual devices in the system.

HMI

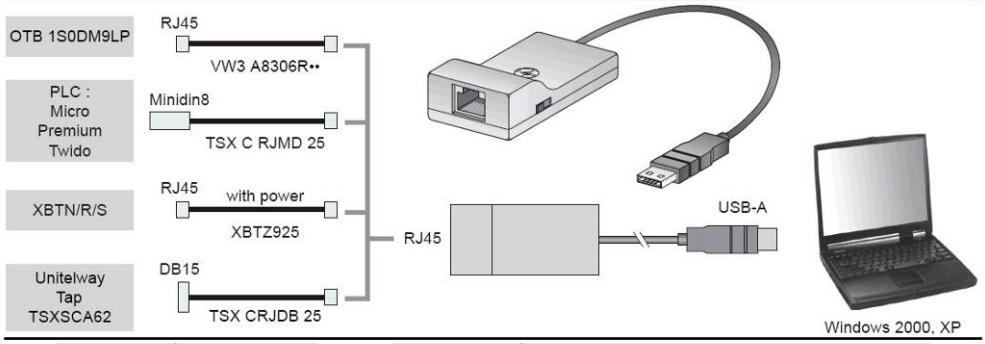
HMI <> PC	XBT RT511 terminals can be connected to the PC in 2 different ways: - via serial port - via USB port
HMI Programming Cable (serial port) **XBTZ945 1/3** **+** **XBTZRT999**	
HMI Programming Cable (USB port) **TSXCUSB485** **&** **XBTZ925**	
Communication Cable **HMI <> Twido** **XBTZ9780**	

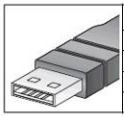
Controller

Communication Cable With USB Connector **PC <> Twido** **TSXCRJMD25**	To program the Twido, use the cable **TSXCRJMD25** with the RS845 adapters (**TSXCUSB485**) to connect the PC with the Twido. Verify that the adapter is in switch position **2.** For the communication to work properly, port 1 must be configured on the Twido

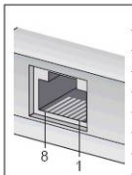
TSX CUSB485
USB to RS485 converter
Quick Reference Guide

TSX CUSB485 Konverter USB-RS485 Kurzanleitung
TSX CUSB485 Convertisseur USB-RS485 Instruction de service
TSX CUSB485 Convertidor USB-RS485 Guía de referencias rápidas
TSX CUSB485 Convertitore USB-RS485 Guida di riferimento
TSX CUSB485 USB 至 RS485 转换器 快速参考指南

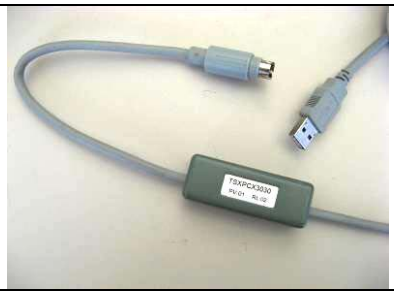




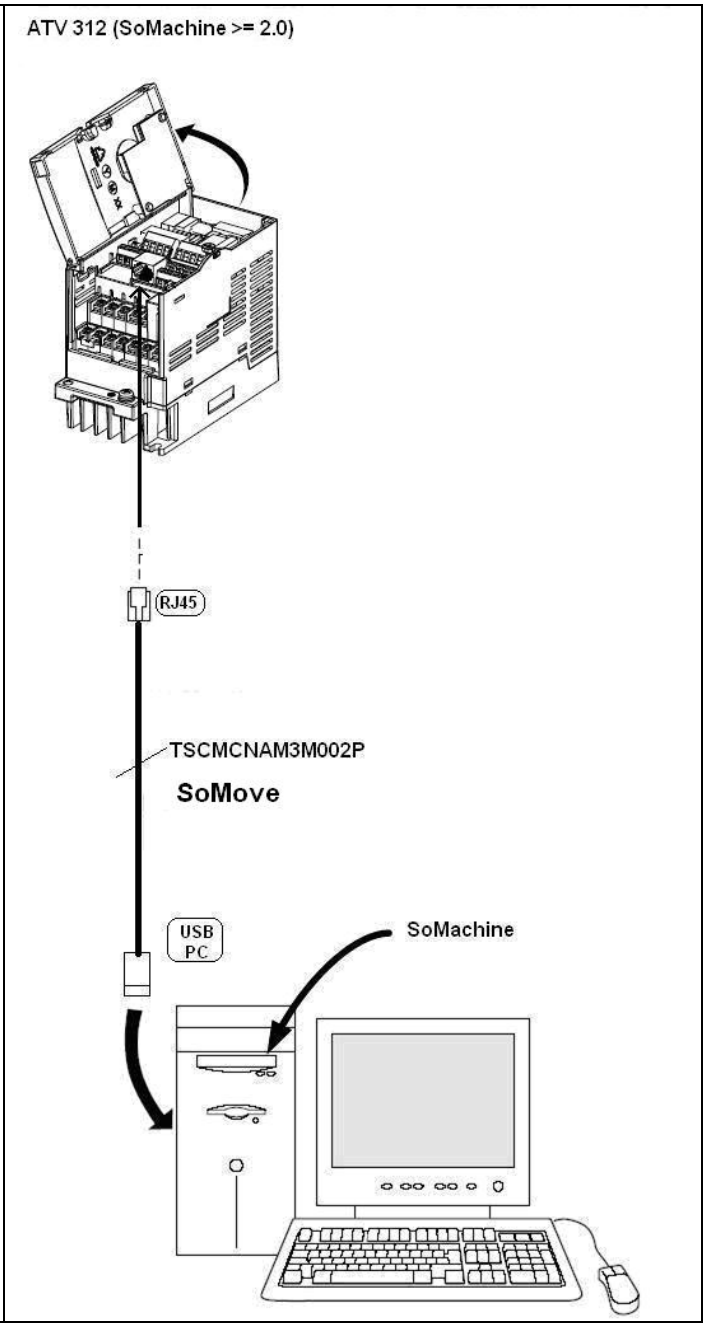
USB-A	
1	VBUS (+5V)
2	Data (D-)
3	Data (D+)
4	U-GND



RJ45		
1	N.C.	
2	N.C.	
3	PMC	Port Mode Control
4	D1 (B)	Transceiver Terminal 1
5	D0 (A)	Transceiver Terminal 0
6	NDE	RS485 Driver Enable (INPUT)
7	VP (OUT)	Power Supply (5V, 500 mA max)
8	Common	Signal and power supply Common

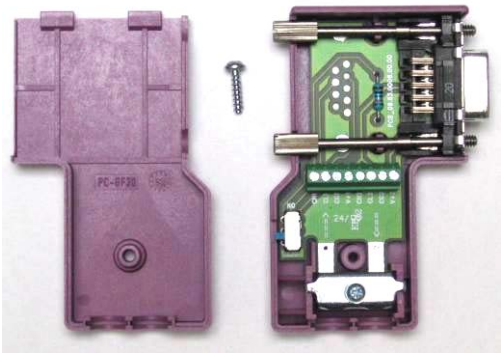
Twido Programming Cable (alternative) **Twido** **TSX PCX 3030** USB Cable	

Altivar 312

Cable set for **SoMove Lite** **TSCMCNAM3M002P** USB cable	

CANopen

CANopen **Master Module** **TWDNCO1M** The CANopen-Interface-Module is plugged into the Twido and has a Sub-D-9 CANopen connection	

CANopen Connector **TSXCANKCDF90T** Use this connector on the **TWDNCO1M**. Connector includes a terminal resistor. CANopen-Cable **TSXCANCD50** Flexible Cable	
CANopen TAP **TSXCANTDM4** Terminal resistor can be selected using the **ON /OFF** switch. The image shows the resistor set to **OFF.**	
CANopen Cable **TCSCCN4F3M1T** Connects from the TAP to the Altivar 312.	

Implementation

Introduction

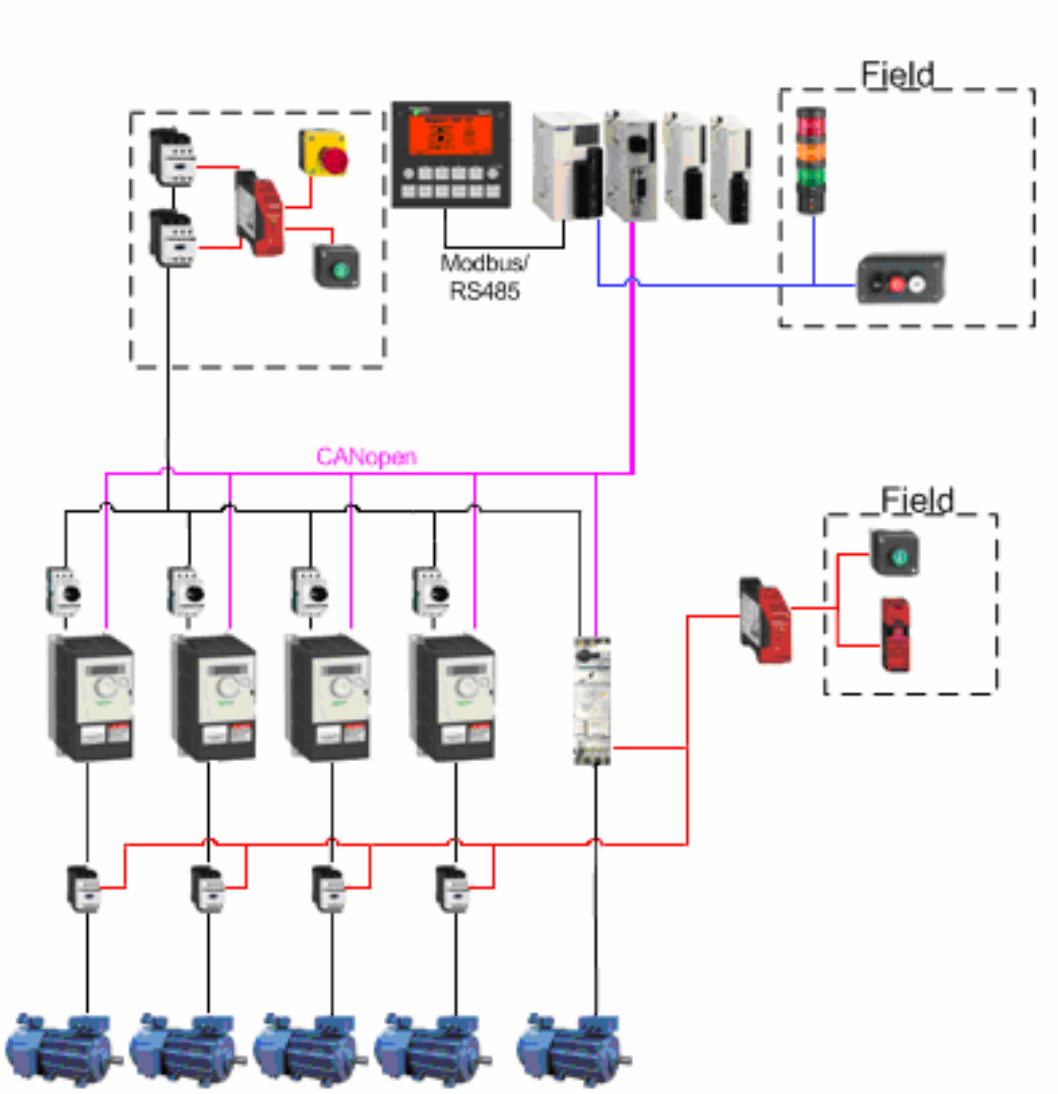
The implementation chapter describes all the steps necessary to initialize, configure, program and start-up the system to achieve the application functions as listed below.

Function

Start-up and functional description

1. Switch on all circuit breakers. The master switch is off.
2. Verify that the Emergency Stop has not been activated
3. Switch on at the master switch
4. Check the door guard switch
5. Acknowledge the Emergency Stop and door guard switch
6. The start-up screen is shown on the HMI touch screen
7. Press start on the display to open the overview screen (the HMI function keys shown below are without function in this sample application)
8. Select the Drive control screen for start / stop, forward / reverse, and speed
9. Use the CANopen status screen to check the status of the devices. Reset resets the CAN communication in the drives
10. Select the Twido status screen to view the status and the controller cycle-time

Functional Layout



Communication

Introduction

This chapter describes the data passed via the communications networks (e.g. CANopen or Modbus) that is not bound directly with digital or analog hardware.

The list contains:

- The device links
- Direction of data flow
- Symbolic name and
- Bus address of the device concerned.

Device Links

This application uses Modbus and CANopen networks.

Modbus connects:
Magelis Panel (master) <> Twido (slave1)

CANopen connects the following devices:
Twido (master) <> 4x Altivar variable speed drives (addresses 1..4)

Datalink Twido << HMI

HMI	
XBTRT511 *(ModBus-Master)*	
Address	**Designation**
%MW12	Drive 1+2 speed (Hz)
%MW14	Drive 3+4 speed (Hz)
%MW18:x1	Drive 1 CANopen alarm
%MW18:x2	Drive 2 CANopen alarm
%MW18:x3	Drive 3 CANopen alarm
%MW18:x4	Drive 4 CANopen alarm
%MW20:X0	Drive 1+2 forward
%MW20:X1	Drive 1+2 reverse
%MW20:X2	Drive 1+2 force speed
%MW20:X3	Drive 1+2 reduce speed
%MW22:X0	Drive 3+4 forward
%MW22:X1	Drive 3+4 reverse
%MW22:X2	Drive 3+4 force speed
%MW22:X3	Drive 3+4 reduce speed

Datalink Twido >> HMI

HMI	
XBTRT511 *(ModBus-Master)*	
Address	**Designation**
%MW12	Motor 1+2 speed
%MW14	Motor 3+4 speed
%MW20:X5	Drive 1+2 reset CANopen communication
%MW22:X5	Drive 3+4 reset CANopen communication
%MW26	Twido status
%MW27	Twido avg. Cycle time

Datalink Twido<< Drive1

Drive 1	
Altivar (*CANopen-slave 1)* **to Twido** *(CANopen-Master)*	
Address	**Designation**
%IWC1.0.0	D_STATUS_SLAVE_1
%IWC1.0.1	D_CONTROL_SLAVE_1
%IWC1.0.2	D_IERROR_SLAVE_1
%IWC1.0.3	D_MCURRENT_SLAVE_1

Twido >> Drive1

Twido *(CANopen-Master)* **to Altivar** (*CANopen-slave 1)*	
Address	**Address**
%QWC1.0.0	D_COMMAND_SLAVE_1
%QWC1.0.1	D_TARGET_SLAVE_1

Datalink Twido << Drive2

Drive 2	
Altivar (*CANopen-slave 2)* **to Twido** *(CANopen-Master)*	
Address	**Designation**
%IWC1.1.0	D_STATUS_SLAVE_2
%IWC1.1.1	D_CONTROL_SLAVE_2
%IWC1.1.2	D_IERROR_SLAVE_2
%IWC1.1.3	D_MCURRENT_SLAVE_2

Twido >> Drive2

Twido *(CANopen-Master)* **to Altivar** (*CANopen-slave 2)*	
Address	**Address**
%QWC1.1.0	D_COMMAND_SLAVE_2
%QWC1.1.1	D_TARGET_SLAVE_2

Datalink Twido << Drive3

Drive 3	
Altivar (*CANopen- slave 3)* **to Twido** *(CANopen-Master)*	
Address	**Designation**
%IWC1.2.0	D_STATUS_SLAVE_3
%IWC1.2.1	D_CONTROL_SLAVE_3
%IWC1.2.2	D_IERROR_SLAVE_3
%IWC1.2.3	D_MCURRENT_SLAVE_3

Twido >> Drive3

Twido *(CANopen-Master)* **to Altivar** (*CANopen- slave 3)*	
Address	**Address**
%QWC1.2.0	D_COMMAND_SLAVE_3
%QWC1.2.1	D_TARGET_SLAVE_3

Datalink Twido << Drive4

Drive 4	
Altivar (*CANopen- slave 4)* **to Twido** *(CANopen-Master)*	
Address	**Designation**
%IWC1.3.0	D_STATUS_SLAVE_4
%IWC1.3.1	D_CONTROL_SLAVE_4
%IWC1.3.2	D_IERROR_SLAVE_4
%IWC1.3.3	D_MCURRENT_SLAVE_4

Twido >> Drive4

Twido *(CANopen-Master)* **to Altivar** (*CANopen- slave 4)*	
Address	**Address**
%QWC1.3.0	D_COMMAND_SLAVE_4
%QWC1.3.1	D_TARGET_SLAVE_4

Controller

Introduction

This chapter describes the steps required for the initialization and configuration and the source program required to fulfill the functions.

The controller program is created using TwidoSuite.

Requirements

To use TwidoSuite and program the Twido first verify that:

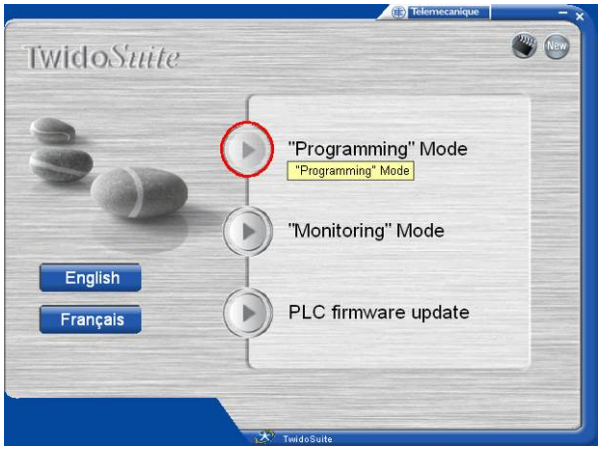
- TwidoSuite is installed on your PC
- The example application, TwidoSuite project **Example.xpr,** is in the standard project directory (C:\Program Files\Schneider Electric\TwidoSuite\My Projects)
- The Twido is switched on
- The Twido is connected to the PC using the proper programming cable

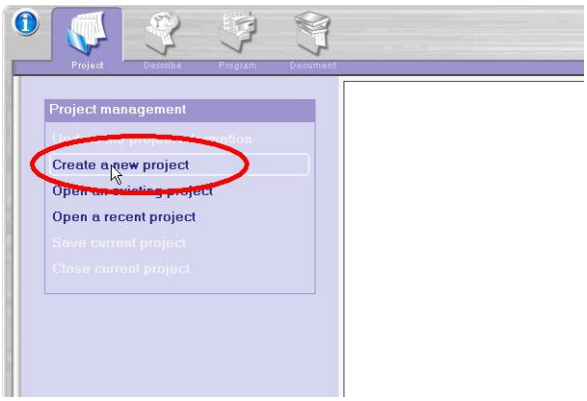
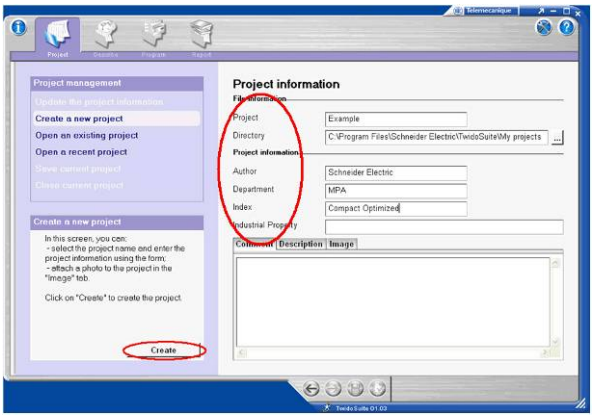
Procedure

Programming the controller consists of:

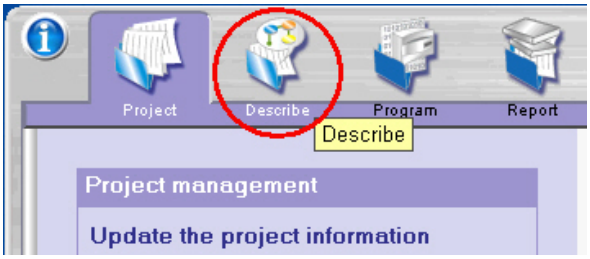
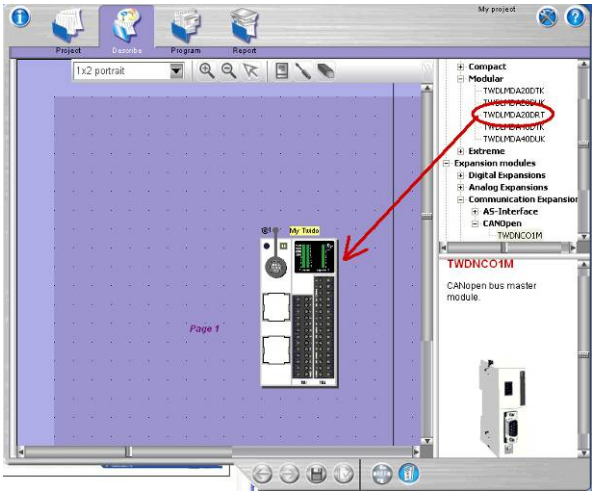
- Create a New Project
- Configure the Twido
- Configure the Modbus
- Configure the CANopen bus
- Activate and define the CANopen slaves
- Create Variables
- Create an Application Program
- Open an existing project
- Download the Program to the Twido
- Save the Program to Memory Card
- Program Duplication with Memory Card

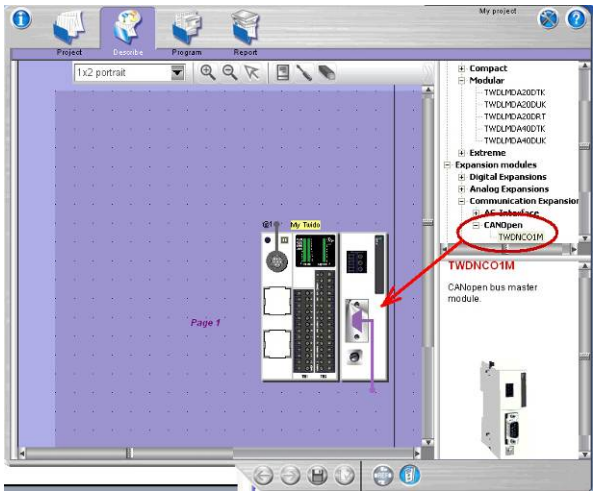
Create a New Project

1	To create a new project click on the arrow next to **''Programming'' Mode**	

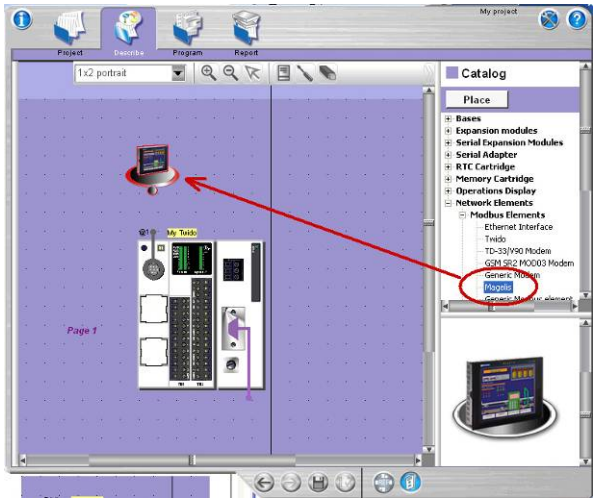
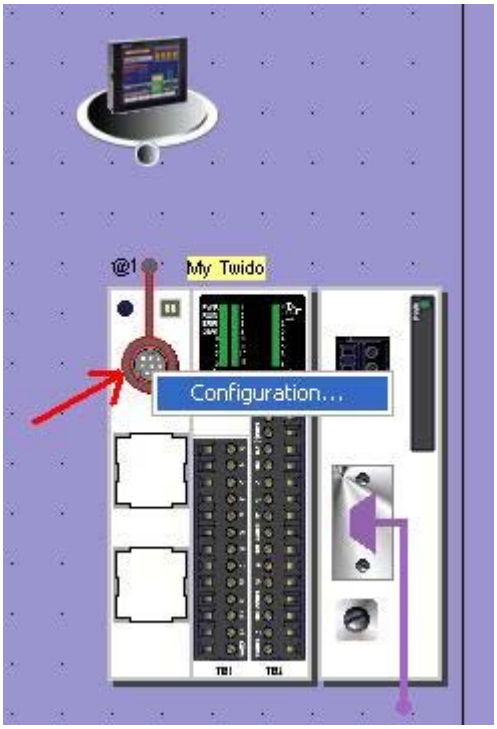
2	Select the **Project** tab and click on: **Create a new project**	
3	Enter the project information as required: • **Project** name • **Directory** path for save • **Author** • **Department** • **Index** • **Industrial Property** Continue with **Create.**	

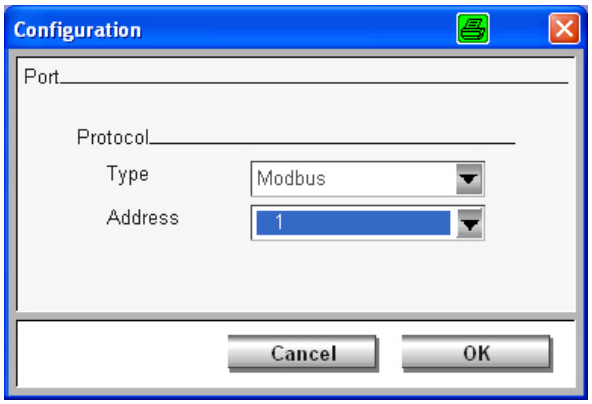
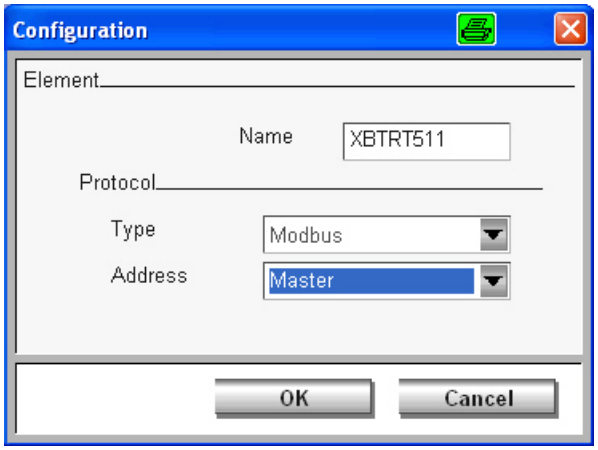
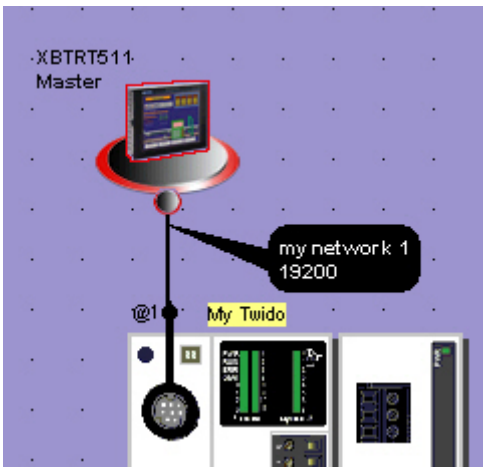
Configure the Twido

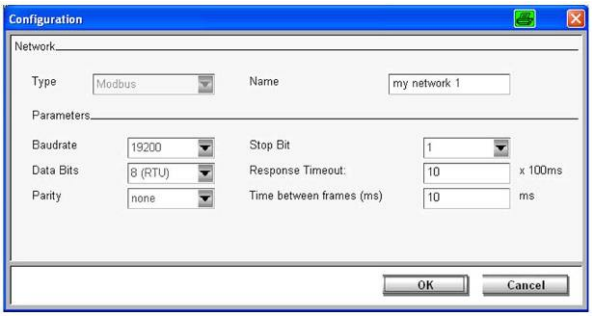
1	To configure the hardware, the communication parameters and bus system, click on the **Describe** tab.	
2	From the **Catalog** on the right select the Twido type: **Bases**->**Modular->** **TWDLMDA20DRT** Click on **Place** to see the Twido in the image of the rack. Pick the Twido from the list and drag it into the page. A green square indicates the possible position.	

3	Again in the **Catalog**, for the communications module, select: **Expansions modules-> Communication Expansion-> CANopen->TWDNCO1M** Click on **Place** and drag the item to the worktop to see the communications module show up in the image of the rack.	

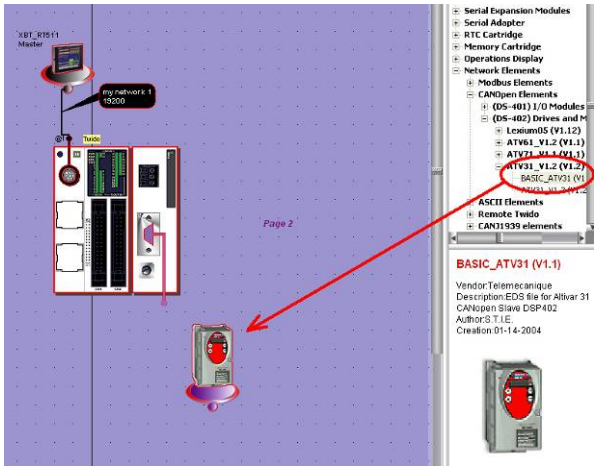
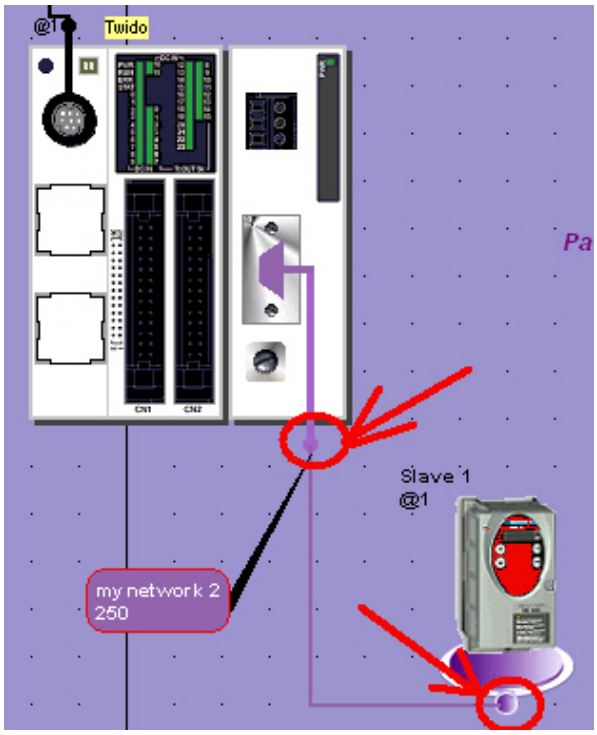
Configure the Modbus

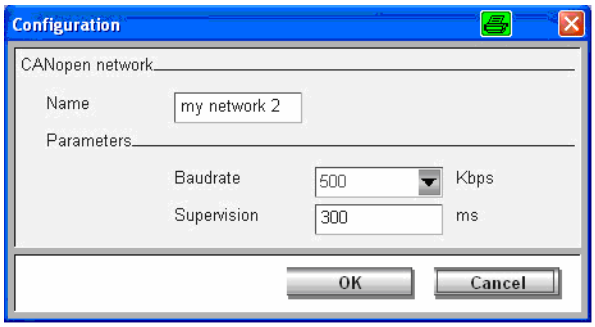
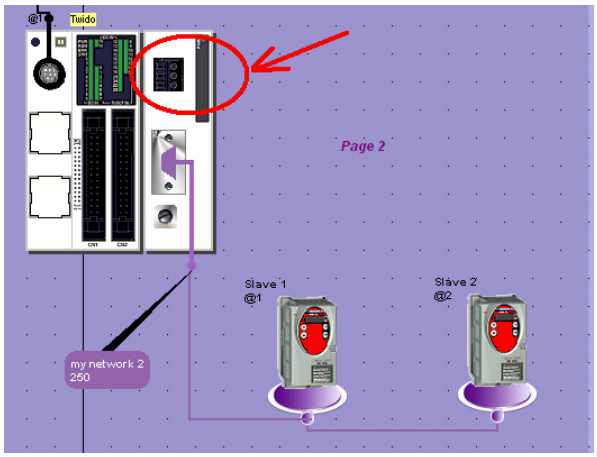
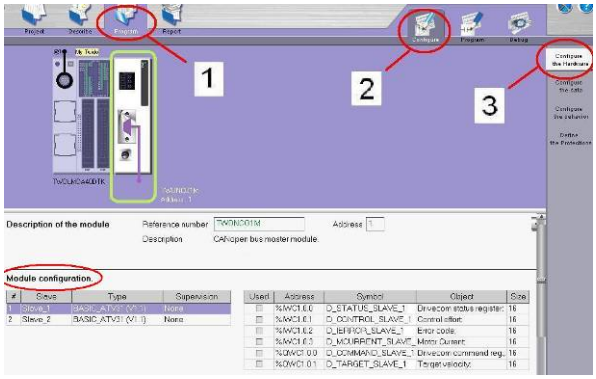
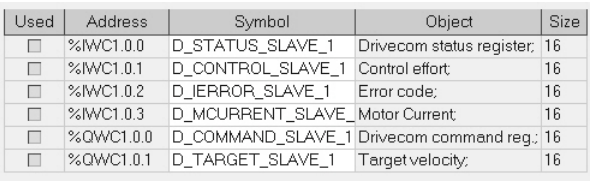
1	Again in the **Catalog**, for the communications module, select: **Network Elements-> Modbus Elements-> Magelis** Pick and place it on the worktop.	
2	To configure the communications port for the HMI connection, right mouse click on the RS485-Port on the image of the Twido and select **Configuration…** in the pop-up menu.	

3	Select: **Type:** Modbus **Address:** 1 Continue with **OK**.	
4	To configure the communication port of the HMI, right mouse click on the HMI image and select **Configuration…** in the pop-up menu.	
5	Insert: **Name:** for example, the HMI type **XBTRT511** **Type: Modbus** **Address: Master** Continue with **OK**.	
6	Now connect the HMI and the Twido with a line. Place the mouse at the end of the Twido port; the view changes. Hold the right mouse button and draw a line to the point below the HMI image. Release the mouse button and the connection, **my network 1** is shown. Double click on: **my network 1** to open the configuration box.	

7	Use the following parameter values: **Baudrate**: **19200** **Data Bits**: **8(RTU)** **Parity**: **none** **Stop Bit**: **1** Confirm with **OK.**	

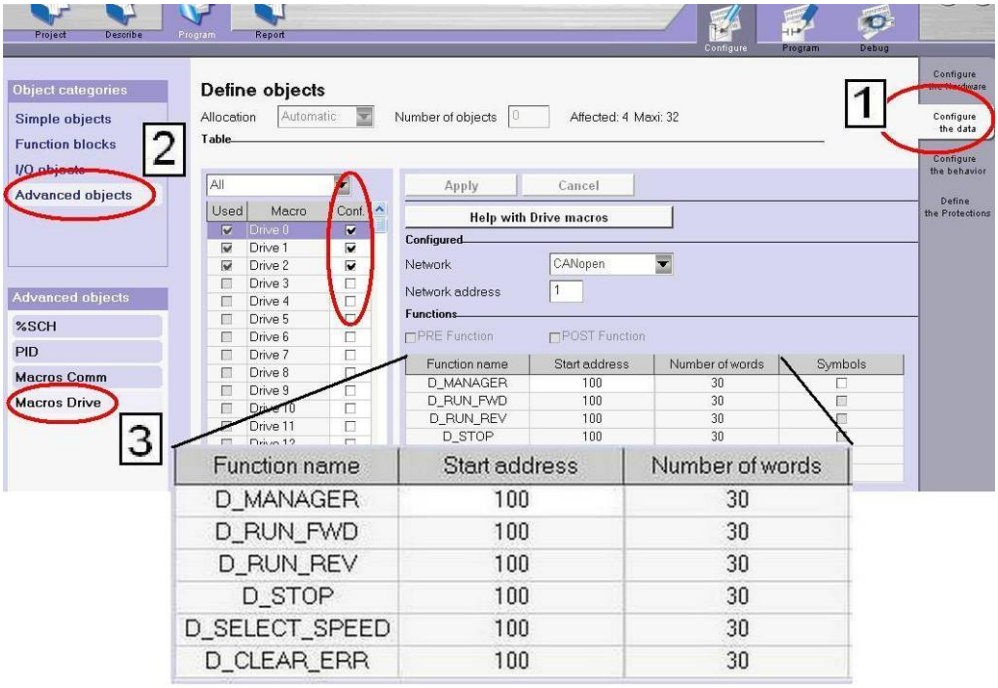
Configure the CANopen

1	First configure the hardware, then activate and define the slaves. Without activation this slave is only shown in the hardware configuration, highlighted in red, but the CANbus communication is inactive. This could be helpful for example, to test a single drive of a machine.	
2	The next step is to add CANopen slaves. **Network Elements->** **CANOpen Elements->** **ATV31_V1.2->** **Basic_ATV31** (the ATV312 is fully compatible with the Altivar 31 regarding to mechanical characteristics and software) Pick and drag it onto the work top.	
3	Connect the CANopen master module and the slave with a line. Place the mouse at the end of the CANopen port; it changes the view. Hold the left mouse button and draw a line to the point below the slave image. Release the mouse button and the connection **my network 2** is shown. Double click on: **my network 2** to open the configuration box.	

4	In this application, the default parameters are used: **Baudrate**: **500** **Supervision: 300** Confirm with **OK.**	
5	Repeat **steps 2 & 3** to add the next ATV312 slave. Connect **slave 1** and **slave 2** with a line. The CANopen communication macros for the slaves will be created automatically. After double clicking the upper half of the CANopen module it opens up the **Module configuration**.	
6	You can go to the **Module configuration** with: **Program->Configure-> Configure the Hardware** **Slave 1** is highlighted in the figure. The table on the right shows the addresses and the descriptions of the default parameters.	
7	The **Used** column indicates whether it is used in the programming code. The **Address** is the variable address in programming code, **Symbol** is the symbolic name and **Object** is a description.	

Used	Address	Symbol	Object	Size
☐	%IWC1.0.0	D_STATUS_SLAVE_1	Drivecom status register;	16
☐	%IWC1.0.1	D_CONTROL_SLAVE_1	Control effort;	16
☐	%IWC1.0.2	D_IERROR_SLAVE_1	Error code;	16
☐	%IWC1.0.3	D_MCURRENT_SLAVE_	Motor Current;	16
☐	%QWC1.0.0	D_COMMAND_SLAVE_1	Drivecom command reg.;	16
☐	%QWC1.0.1	D_TARGET_SLAVE_1	Target velocity;	16

Activate and define the CANopen slaves

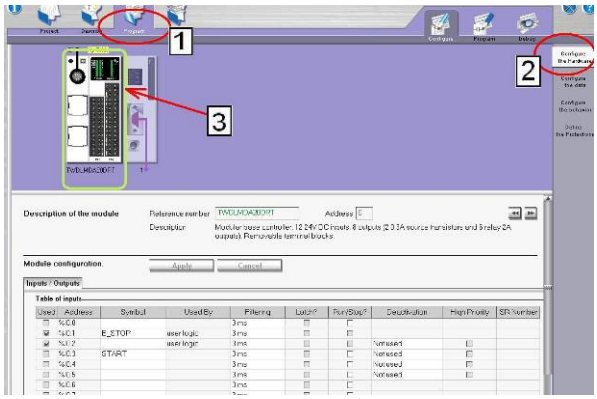
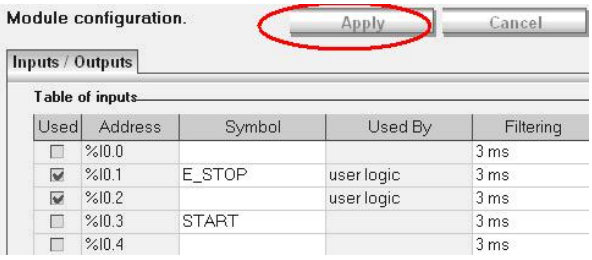
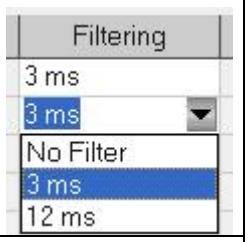
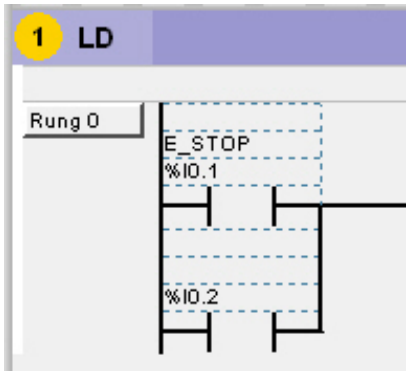
1	Click on: **Configure the data ->Advanced Objects -> Macros Drive** Activate the slave in the column **Conf**. and select CANopen for a **Network**. Select the **Network Address**. In the column **Start Address** enter for example, 100 and for the second slave 200. Confirm with Enter. The ranges from 100 to 130 and 200 to 230 are now reserved for the two slaves. 
2	The slaves are now configured and ready for use in the programming code.

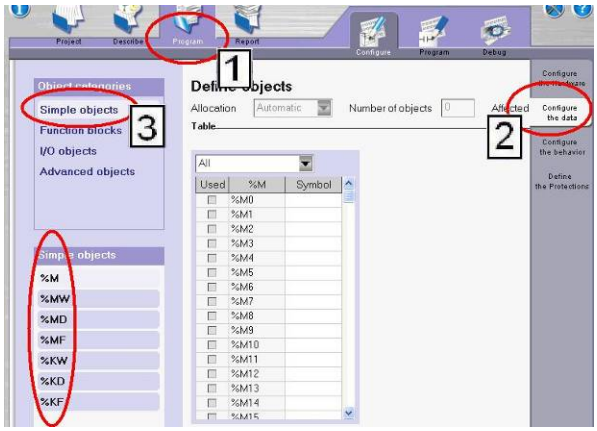
Create Variables

Two different variable types are possible:

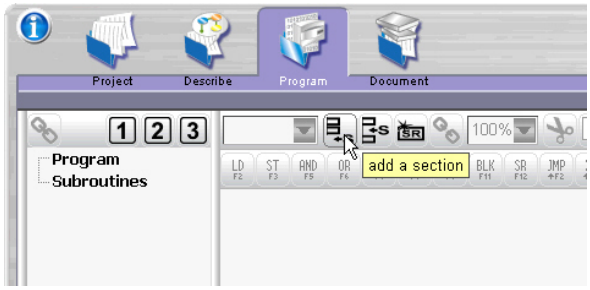
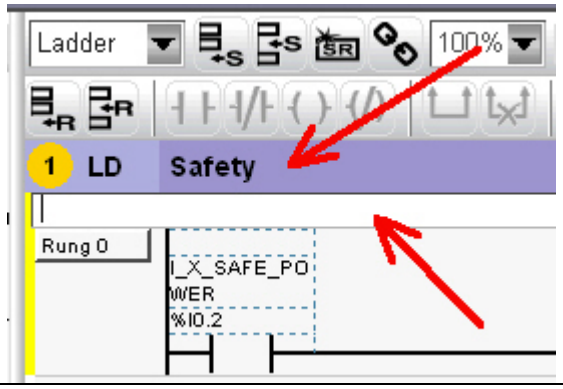
- I/O variables, linked to inputs or outputs
- internal variables, such as bit, byte, word, double word, floating point and constant

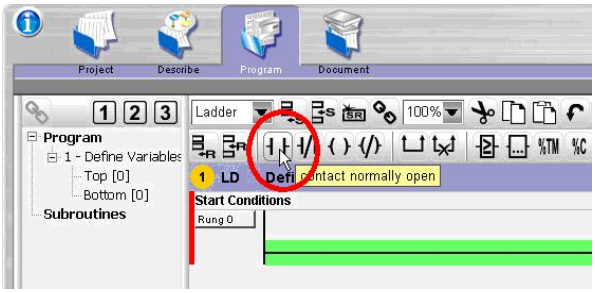
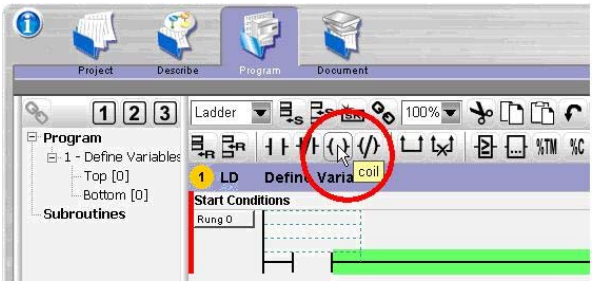
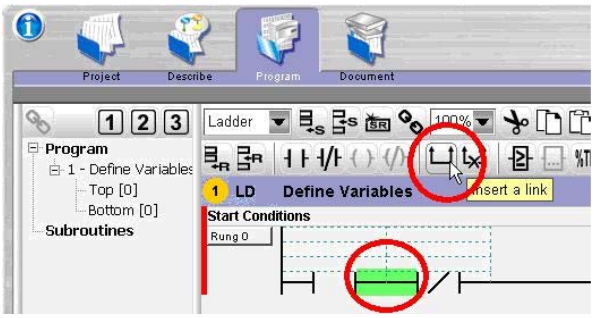
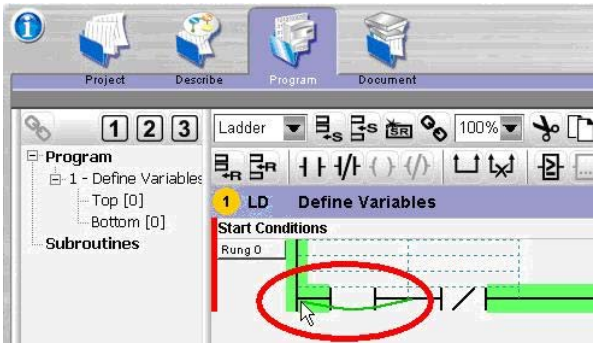
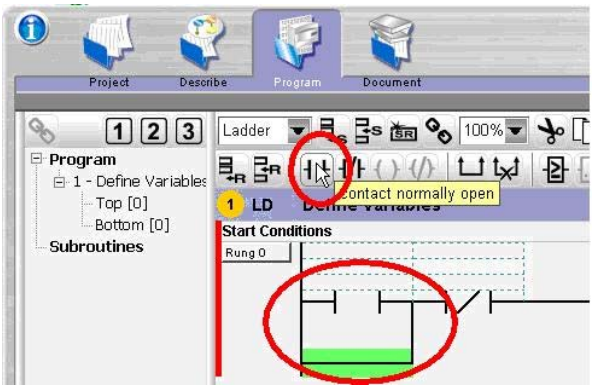
The variables can be defined in separate lists or entered directly in the programming editor (list or ladder).

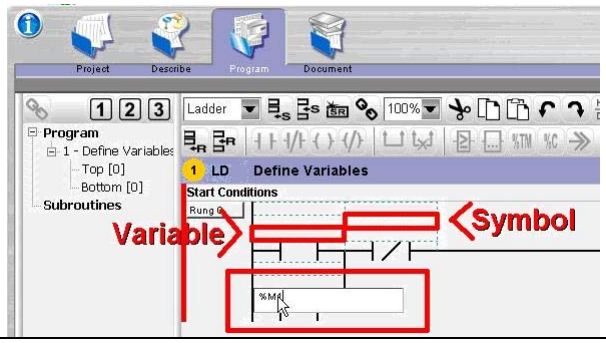
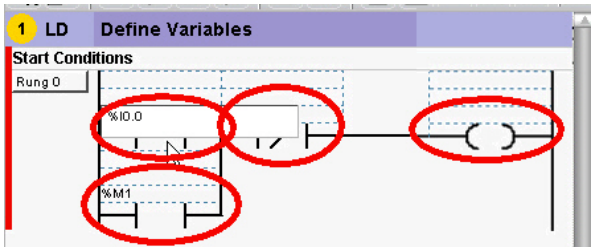
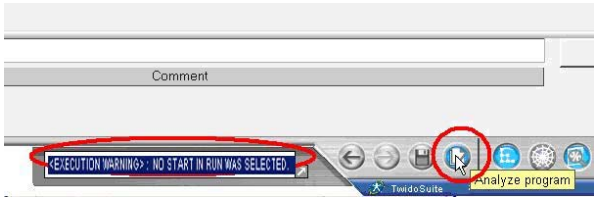
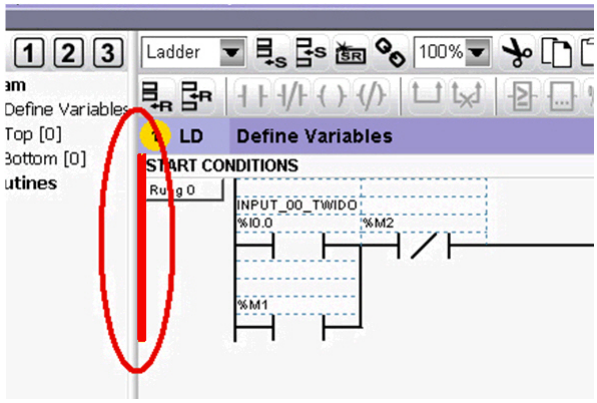
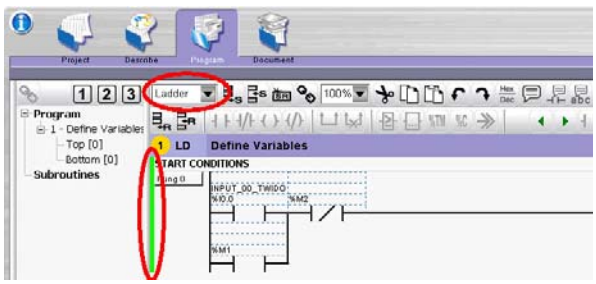
1	To create I/O variables, open the **Module configuration** with **Program**->**Configure the Hardware** Then click the relevant I/O module **(3)**. The table at the bottom of the screen shows the I/O variables.	
2	The following is shown in the table: **Used**: indicates whether the variable is used in the programming code **Address**: the I/O address **Symbol**: the symbolic variable **Used By**: user logic or empty **Filtering**: minimum signal time length (only for inputs) Confirm with **Apply.**	 
3	The image shows how the variables in the table above are used in the program: **%I0.1 – E_STOP:** variable and symbolic name in program code **%I0.2:** variable without symbolic name in program code **%I0.3 - START:** the symbolic name is in the variable list (see image above) but not used in the program.	

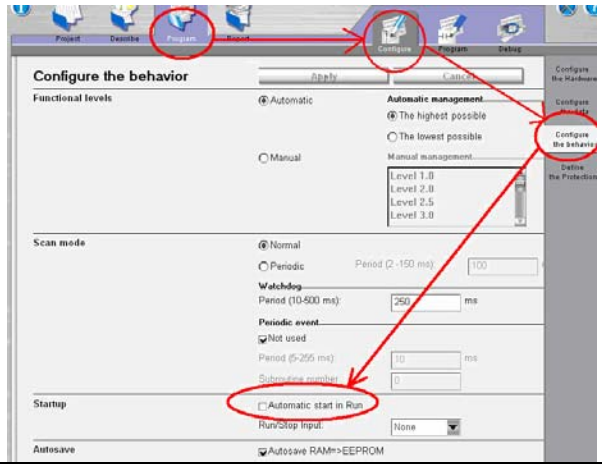
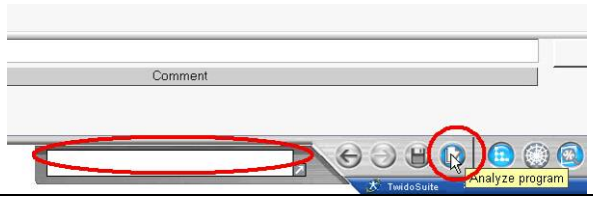
4	To create internal variables, open the Module configuration: **Program->** **Configure the Data->** **Simple Objects** The table, bottom left, lists the datatypes: **%M, %MW, %MD**, etc. A table with addresses and symbolic names for each data type is shown on the right. **Used**: indicates whether the variable is used in the program **%Mx:** variable address **Symbol**: symbolic variable	

Create a Program

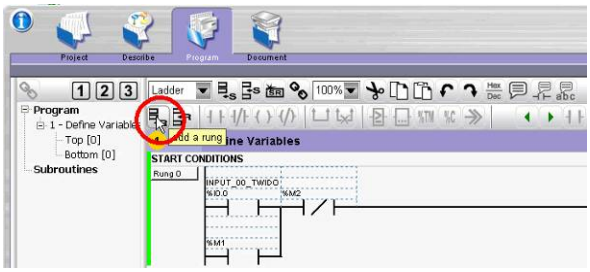
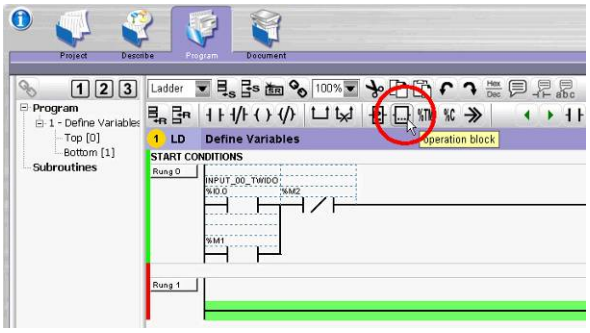
1	Go to the program editor with: **Program->Edit Program** A new program always begins with **LADDER** as the programming language. You can only view the program in LIST mode once the first rung has been completed and analyzed.	
2	Start the programming with: **add a section**. This creates a section with empty rungs.	
3	Click on the header of a **Section** to enter a description (for example "**Safety"**). Click on the header of a rung to enter a description.	

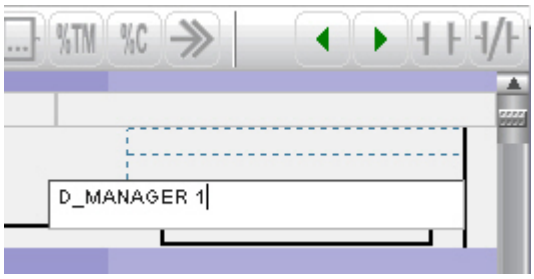
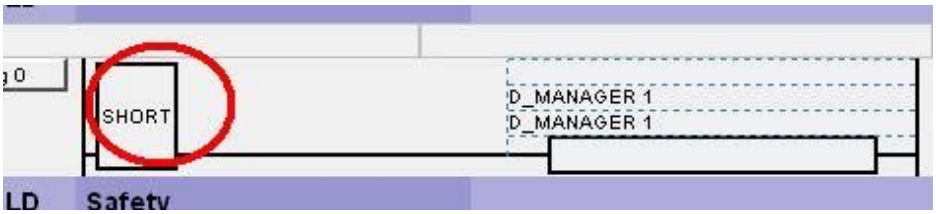
5	To add contacts, click on the rung. The rung changes its color and indicates the current rung is being edited. Clicking on an icon in the toolbar inserts the selected object at the first available place on the rung.	
6	Add coils in the same manner.	
7	To make a logical connection, simply mark the source and pull it to the destination. All possible connecting points are shown in same color.	
8	Once a link has been inserted, you can add a contact to form a logical OR.	
9	To assign an object to an address or variable, click on the upper half of the object.	

10	The editor automatically recognizes whether a rung is syntactically complete and marks it in yellow. The rung, however, has not yet been analyzed.	
11	The analysis is done for all contacts.	
12	Analyze the program code by clicking on the icon in the bottom right hand corner of the window. The message box to the left of the icon shows if there are any compilation errors.	 
13	If the analysis is successful, the red band changes to green.	

14	If compilation errors are found, they are listed at the bottom of the edit field. Click on the arrow icon on the right side of the message box to enlarge it. A common compilation message may be: **No Start In Run Was Selected**	
15	To fix this, select: **Program->** **Configure->** **Configure the behavior ->** and for **Startup** activate: **Automatic start in Run** This is an operation mode that causes the Twido to start up automatically on return of power after a power interruption.	
16	After restarting the analysis, the notification field is empty.	

Macro Drive

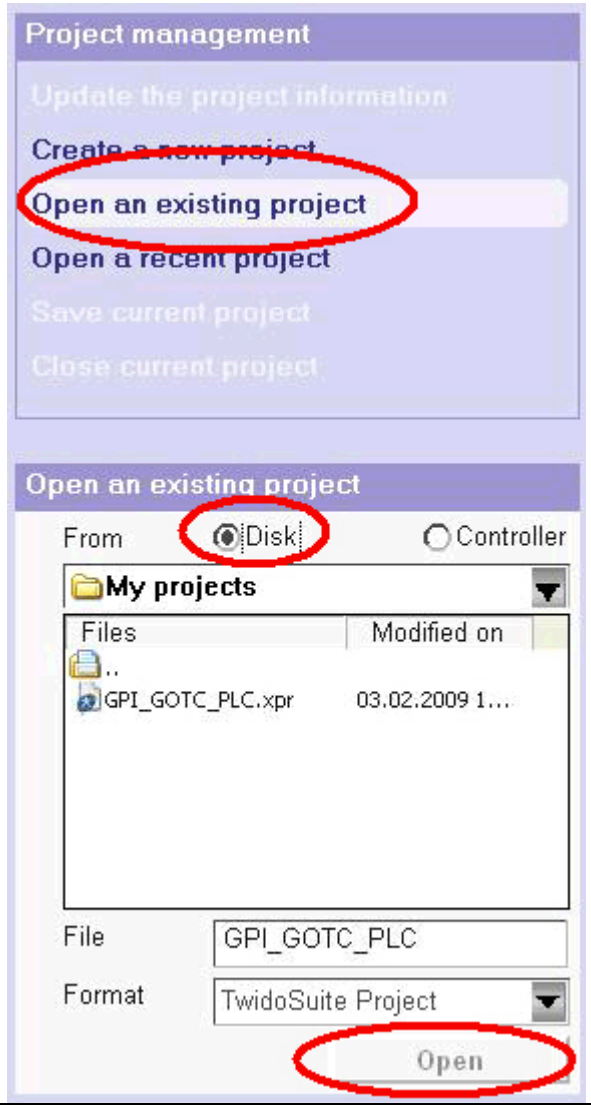
1	To use the macros for the drives, a second Rung is added using the **add a rung** function.	
2	Insert an operation block for the macro.	

3	To link the operation block to the macro drive input **D_manager 1**. D_manager is the macro function to access the drive and the 1 indicates Drive 1. The space between D_manager and the 1 is mandatory. Since the macro is already configured, the address field also contains the symbolic macro name **D_MANAGER 1**. The **SHORT** (see next step) must be exchanged for a control variable. You can now repeat this for the other macros: **D_CLEAR_ERR** **D_RUN_FWD** **D_RUN_REV** **D_STOP** **D_SELECT_SPEED**	
4		

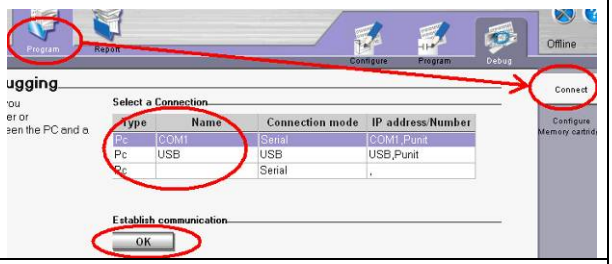
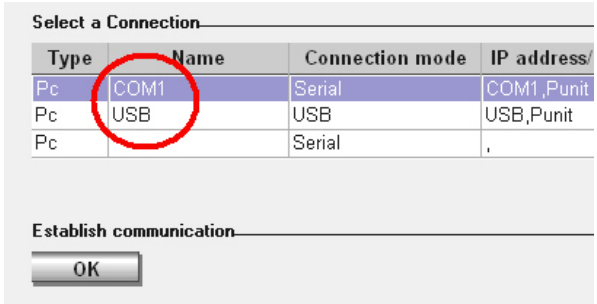
Save / Export the project

1	To save the project select **Save current project** in the project menu. In the **Save current project** window you can select the folder for the save, input the **File** name and select the file **Format**. You can save the file as a normal project file or as an archive file. The archive file has the advantage of being portable and is a single file.	

Open / Import a Project

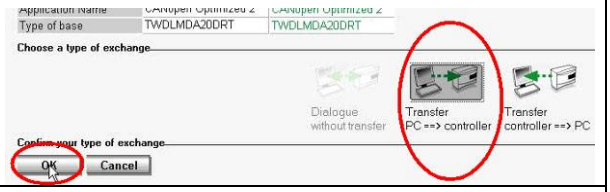
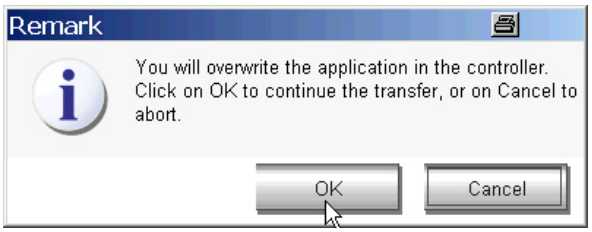
1	To open a project use the **Open an existing project** function in the project menu. In the **open an existing project** window select the medium: **Disk** or **Controller** by clicking the radio button. Choose the file type under **Format (**for a project file: **Project)**, select the folder and select the **File** you wish to open. When ready click on **Open** to open the project For import the action is the same, except the file **Format** is TwidoSuite Archive.	

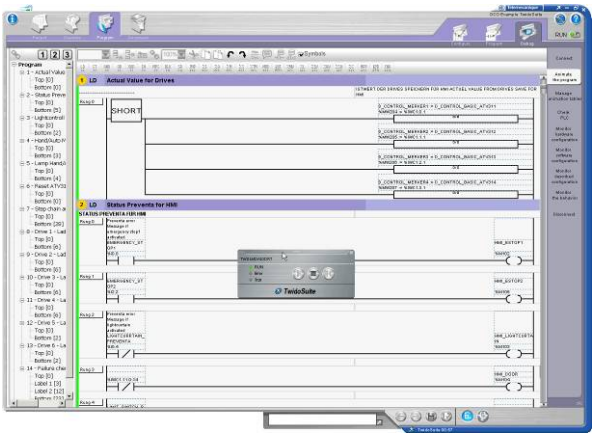
Connecting to the Twido

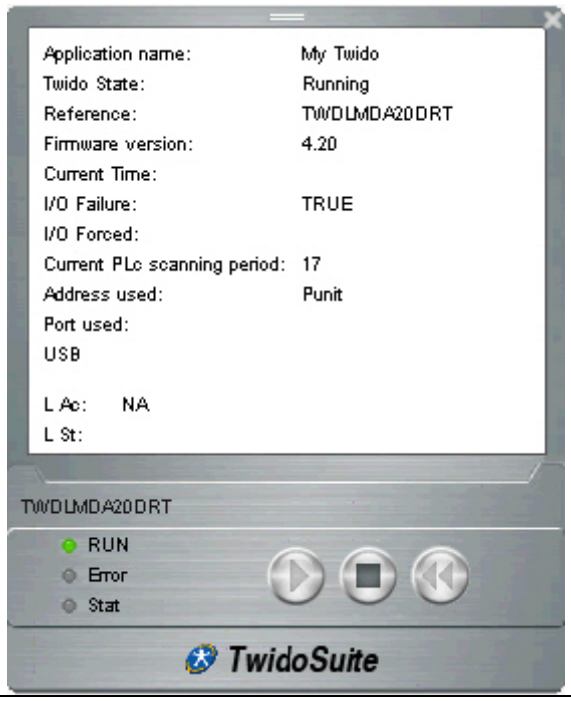
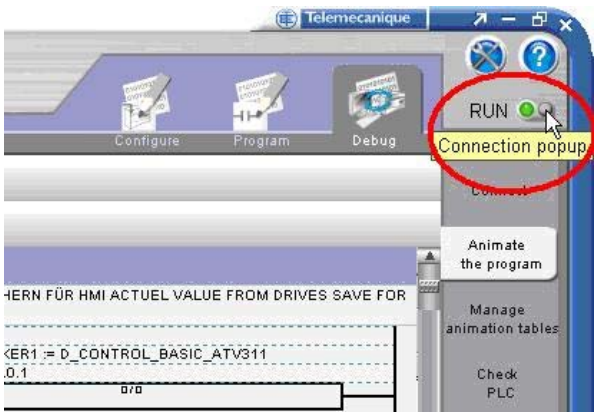
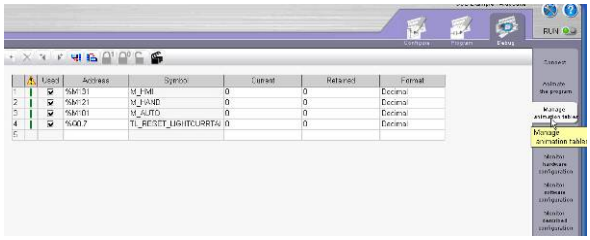
1	To connect the PC to the Twido select **Program** in the main menu, then **Debug** in the sub-menu to get access to the connect selection.	
2	You can now either configure a new connection (see menu at bottom of window) or select a pre-configured connection. Select serial or USB. Connect with **OK**.	

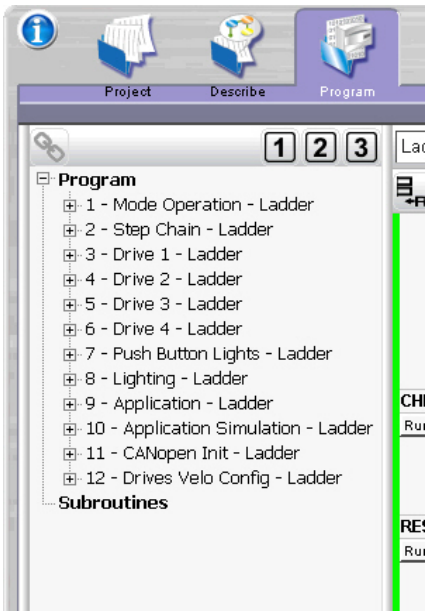
3	A progress bar is displayed while TwidoSuite establishes the connection.	

Download a Program

1	After successfully connecting to the Twido, TwidoSuite compares the project in the Twido and opened project. You can now either download your project or upload the contents of the Twido	
2	To download the project select: **Transfer PC ==> controller** and confirm with **OK.**	
3	If the Twido is already running it must be stopped first. Confirm with **OK** if you wish to proceed.	
4	You will be informed that the contents of the Twido are about to be overwritten. Confirm with **OK** to continue.	
5	The download status is indicated with a process bar.	

6	After a successful download, TwidoSuite changes to online viewing mode.	
7	A control panel is shown so you can start and stop the Twido. The upper button extends the window, offering more Twido status information.	
8	If you start the Twido, the action must be confirmed. Click **OK** to proceed.	
9	After successfully starting the Twido, the **RUN** LED changes from yellow to green.	

10	The image on the right shows the expanded control panel with the Twido in run mode.	
11	If the control panel is closed, a click in the upper left hand corner of the main window re-opens it.	
12	The sub-menu **Manage animation tables** allows access to the animation tables.	

13	Use the project browser to navigate through the program sections.	

HMI

Introduction

In this application a compact Magelis XBTR511 semigraphic touch-display is implemented. The display communicates via Modbus protocol with the Twido.

Configuration and programming of the display is done using the Vijeo-Designer Lite HMI software.

Requirements

Before proceeding with the configuration verify that:

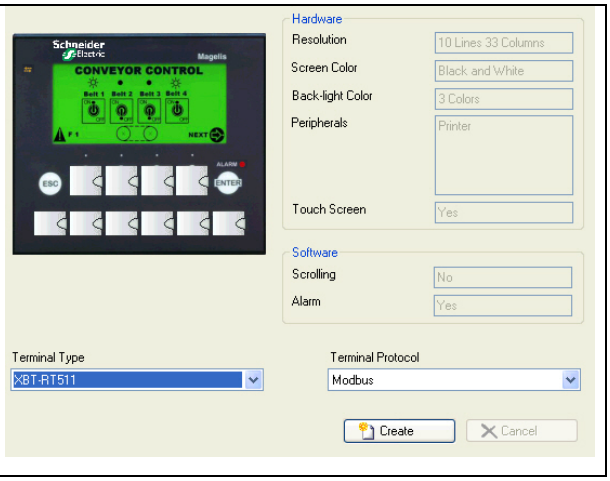
- Vijeo-Designer Lite software is installed on your PC
- The project ”Example.dop“ is available in the directory path (C:\Programs\Schneider Electric\VIJEO-DESIGNER LITE\Apps)
- The Magelis HMI is turned on
- The Magelis HMI and is connected to the PC using cable XBTZ915

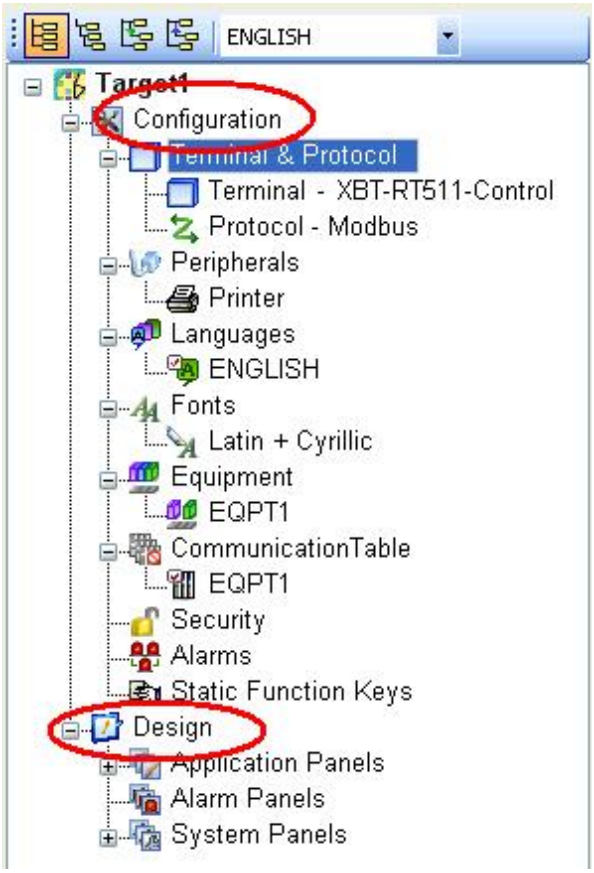
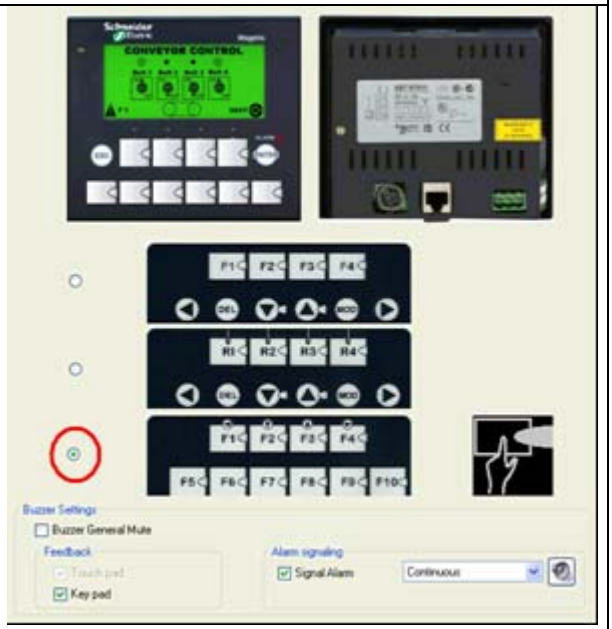
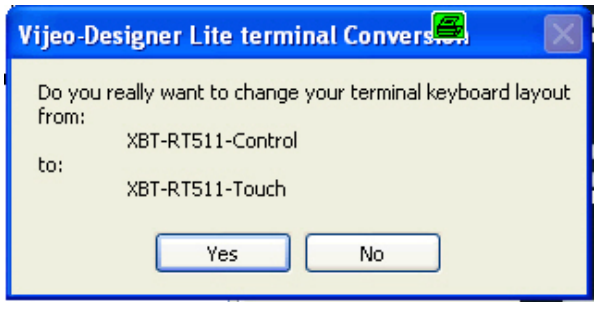
Procedure

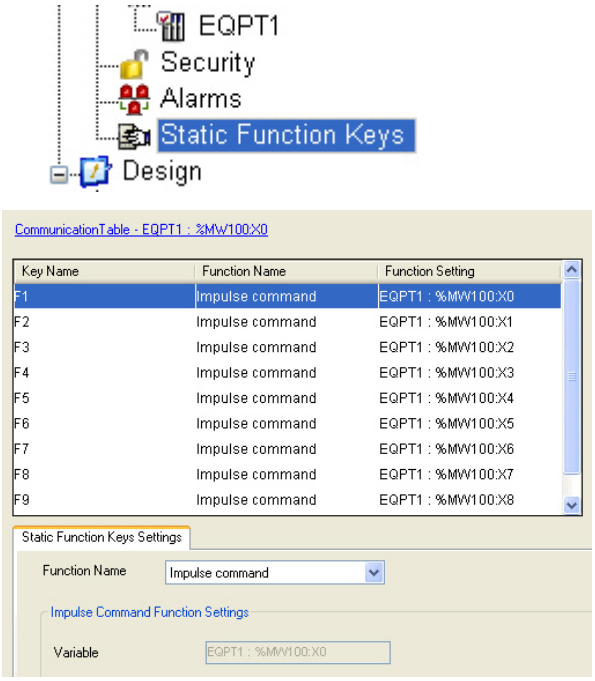
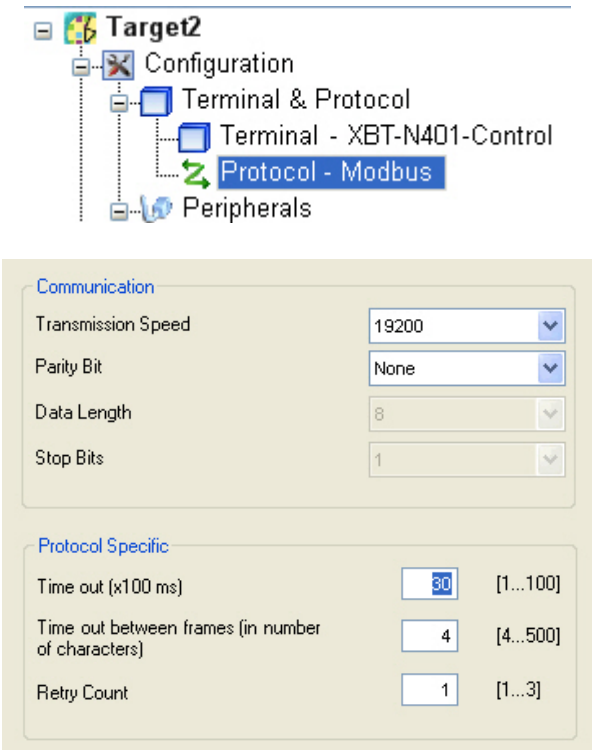
Setting up the HMI is done as follows:

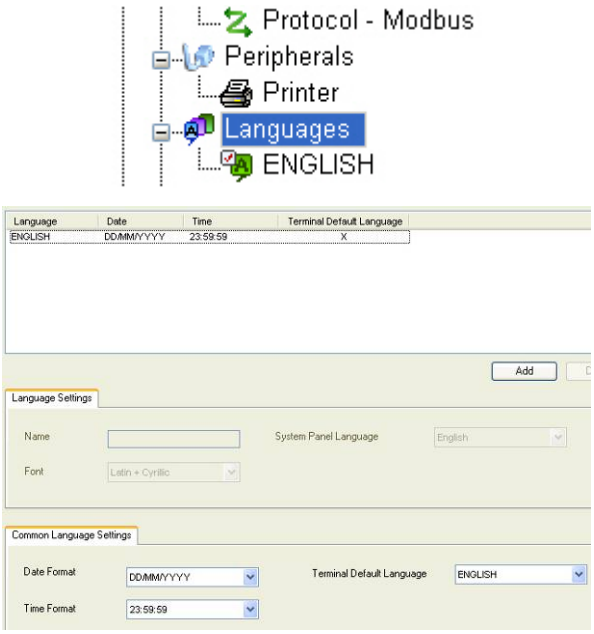
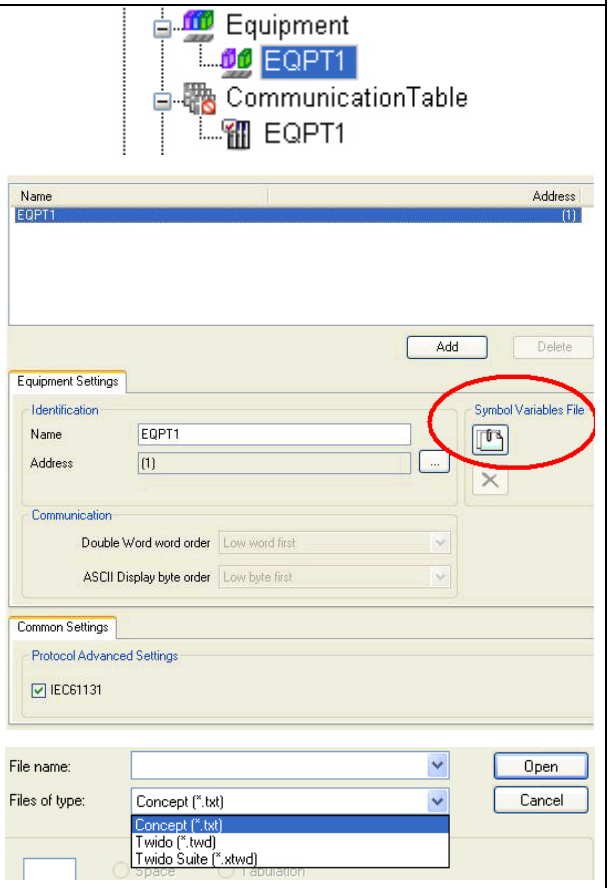
- Create a new configuration
- Create alarm-messages
- Create an application (Design)
- Creation of variables
- Screen Navigation & Viewing System Pages
- Open an existing project and transfer to the HMI
- Driver-Manager: add an interface

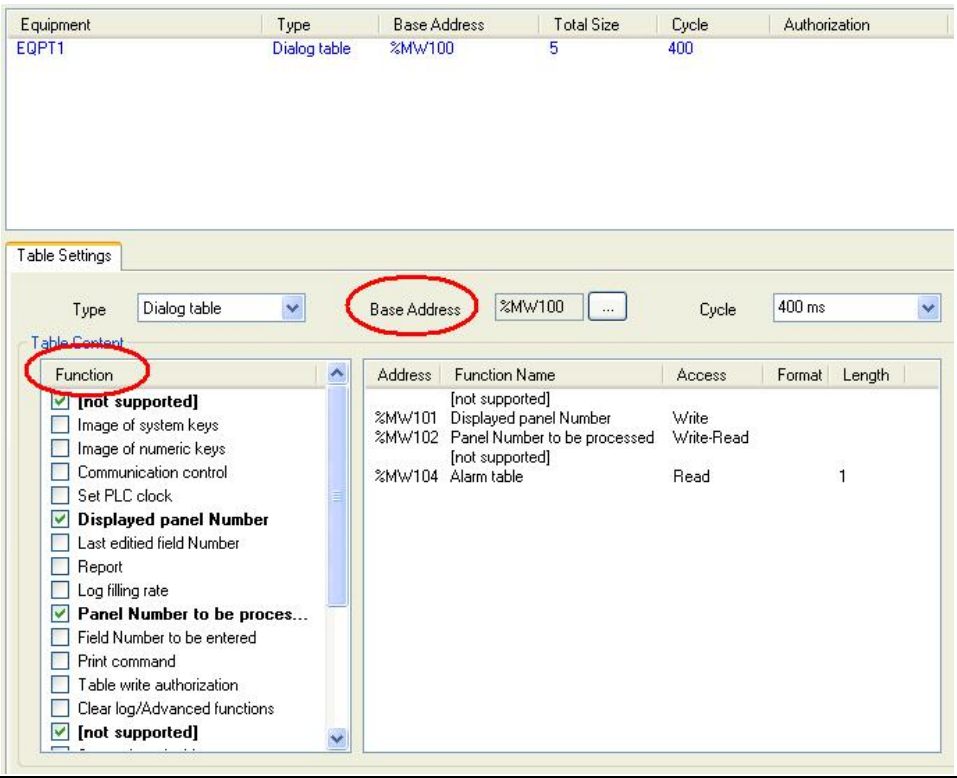
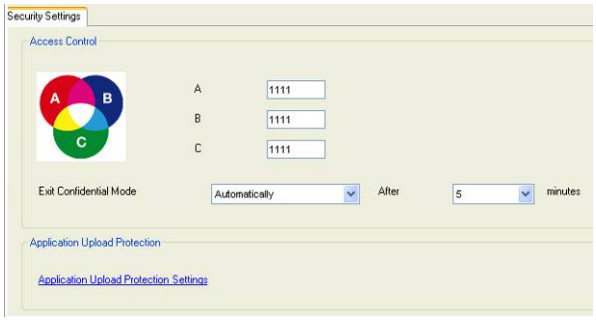
Create a new configuration

1	As **Terminal Type** select **XBTRT511.** As **Terminal Protocol** select **Modbus**. Click on **Create.**	

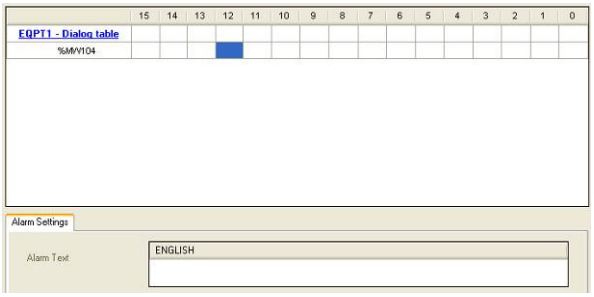
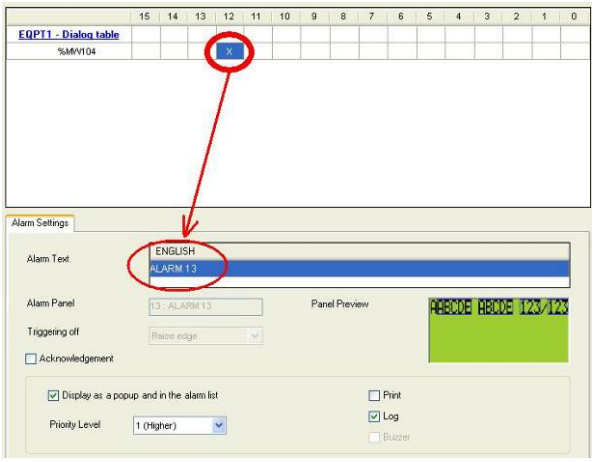
2	On the left hand side of the window you will see the project structure in the browser. The structure is split in two sections: **Configuration** and **Design**. The **Configuration** lists the settings for the basic parameters of the HMI. The **Design** area lists the application and system screens. Click on the **Terminal – XBTRT511- Control** entry, to set up the HMI operational mode.	
3	In this example, the function key option is used. The operator navigation is now available via touch-display. Select the function key option (marked with a red circle). A message box asks you to confirm the change. Click on **Yes.**	 

4	In **Static Function Keys** area, the settings of the F1 to F10 keys can be configured. These are on the front of the HMI. The column **Function Setting** lists the Twido addresses. In the dialog for the **Function Name** the characteristic (impulse or toggle) can be chosen. The function keys can also be configured to switch between two application screens.	
5	Click on **Protocol – Modbus** in the browser to set up the Modbus RTU via RS485 communication. Use the following settings: **Transmission Speed**:**19200** **Parity bit**: **None** **Data Length**: **8** **Stop bits**: **1** The default parameters under **Protocol Specific** can be left as they are.	

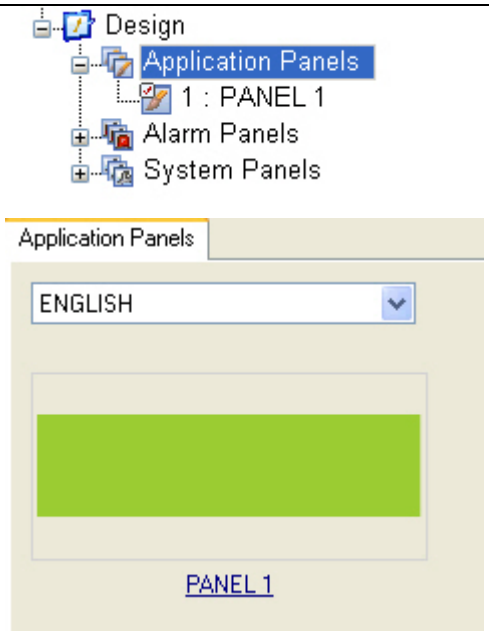
6	It is possible to add languages for multi-lingual operator usage. The **Languages** entry lists the languages offered to the operator at power up.	
7	Under **Equipment** the bus-nodes must be defined. **EQPT1** is a default, but the **Name** can be changed. The corresponding slave address is shown in the **Address** field. Additional slaves can be created using the **Add** button. Use **Symbol Variables File** to import an existing variable list from a Twido program.	
8	The **Communication Table** defines the data structures used in communication, for example, memory buffers, alarm lists, function key definitions, communications control, etc. Right mouse click on **Communication Table** and uncheck **Disabled** in the pop-up menu**.**	

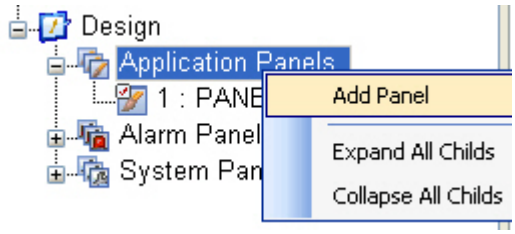
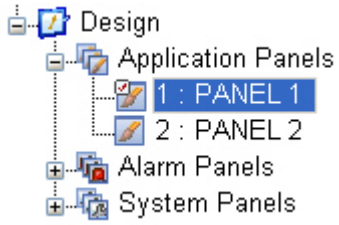
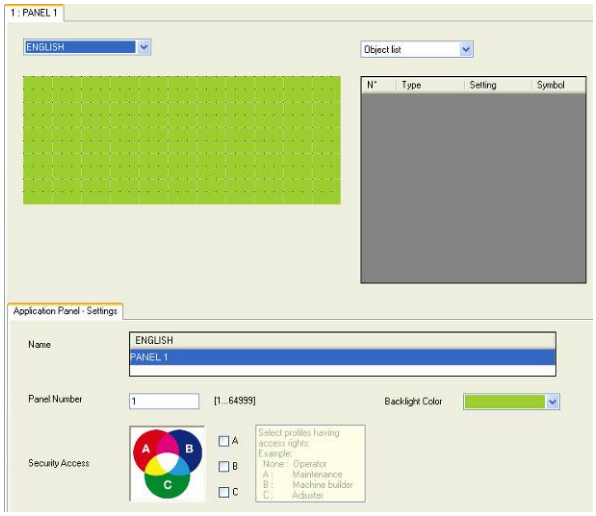
9	Click on the corresponding table **EPQT1**	
10	The **Function** list indicates the activated functions for this equipment. To add more functions**,** activate the checkbox next to the function. Updated selections are listed in the table on the right. The **Base Address** is the start address of the **Dialog table**. The start address and length of each function is shown in the table on the right. **Cycle** defines the cycle time for updating this data. 	
11	Use the **Security** entry to define passwords to protect the HMI from unauthorized use. You can define a time limit for the authorized access in **minutes.** The validity of the password for the access levels (**A**, **B** and **C**) is configurable for individual screens.	 

Create Alarm Messages

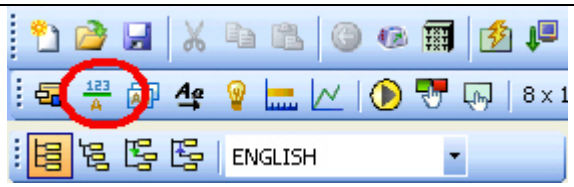
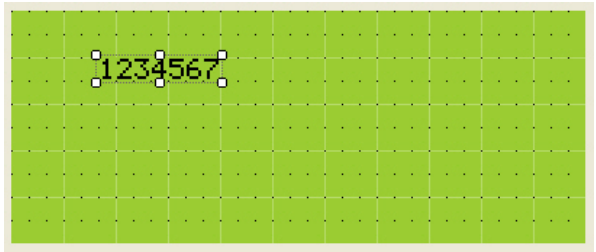
1	Click on **Alarms** to open the alarm-list. This function must first be configured in the **Communication Table** (see above).	
2	The left column shows the address (for example, %MW104).The row shows the individual bits for this memory word. Select a bit by double clicking on the cell.	
3	On the **Alarm Settings** tab add the text to be displayed for this alarm in the **Alarm Text** box. Confirm with Enter. The text is displayed on the screen when an alarm of this type appears. The position of the text on the screen can be defined under **Alarm Panel** and viewed in the **Panel Preview**.	

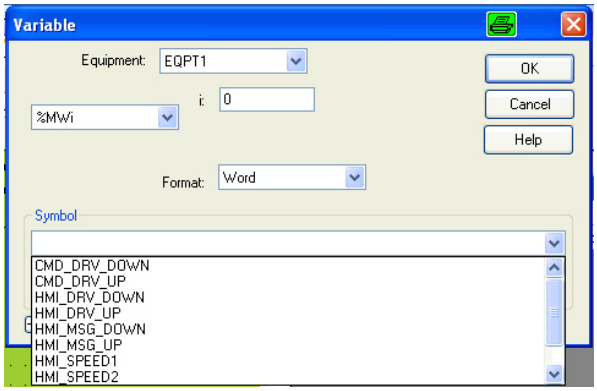
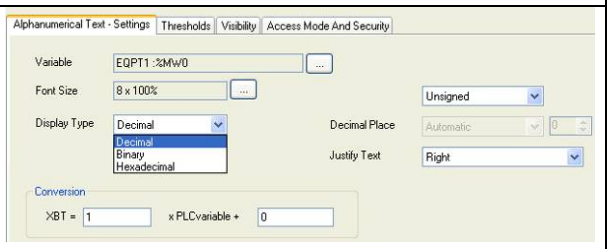
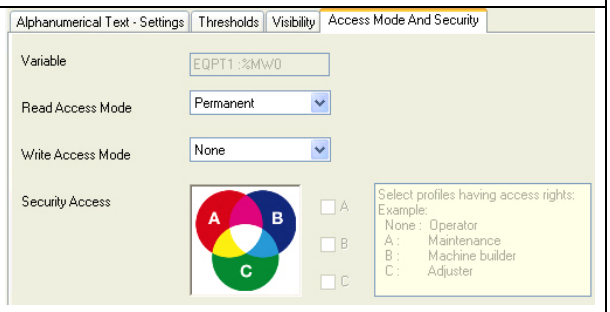
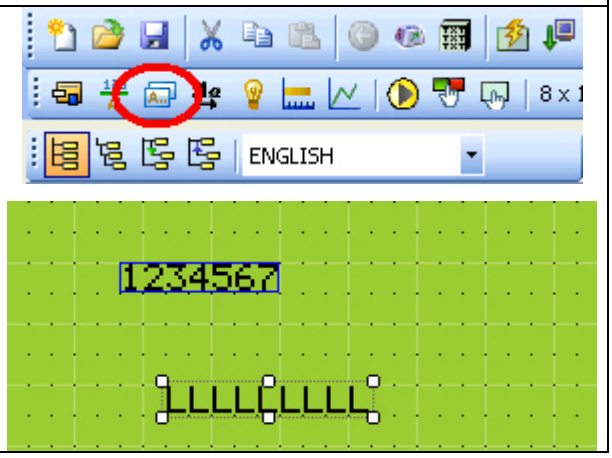
Create an Application screen

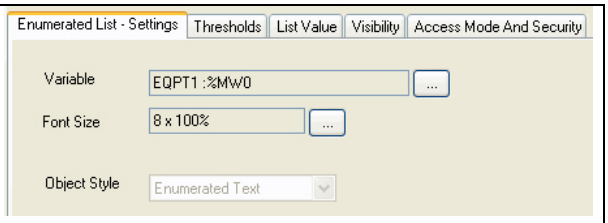
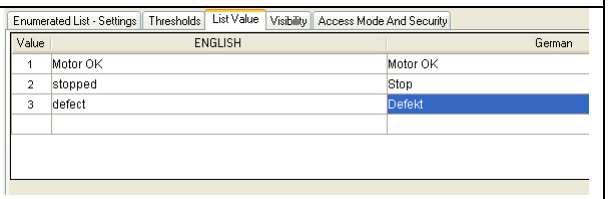
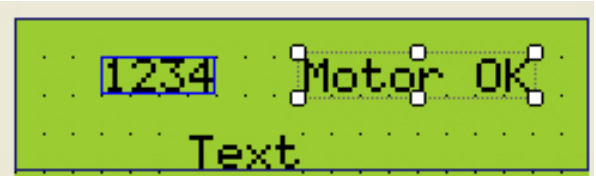
1	Design HMI screens to display system variables using **Application Panels** entry. **PANEL 1** is the default name for the first screen.	

2	To create a new screen, select: **Application Panels**->**Add Panel** in the project browser	
3	Click on the newly created screen name in the project browser to display the actual screen. You can now define the layout of the screen. Add texts wherever you wish to position them**.** The **Security Access** is used to apply the password level required to access this screen.	 

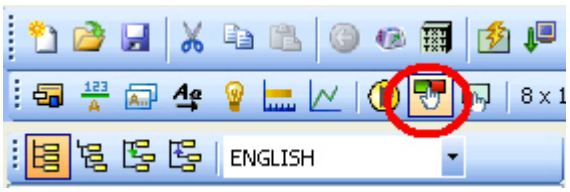
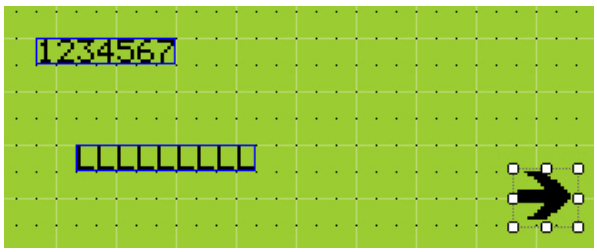
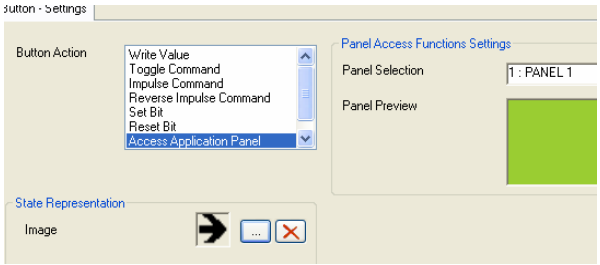
Creation of variables

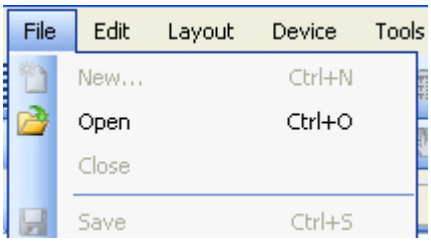
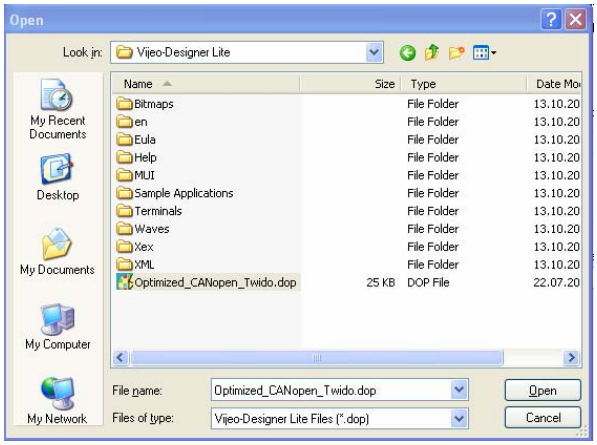
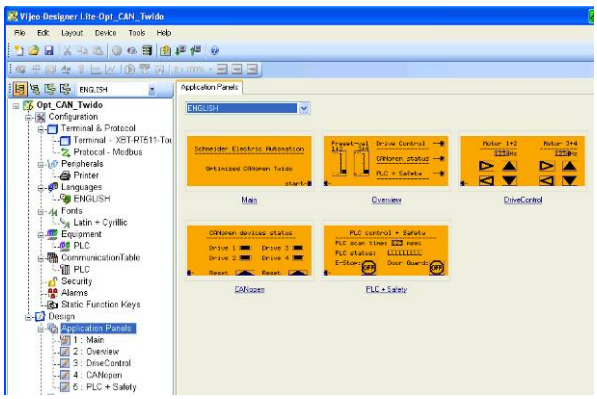
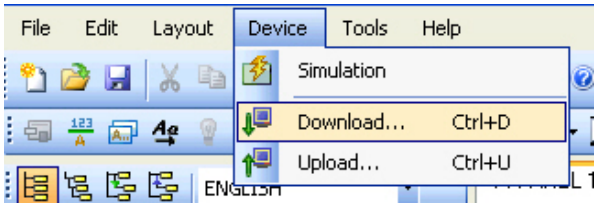
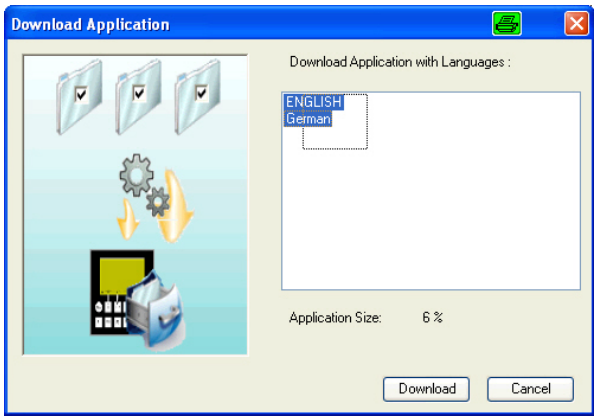
1	For alphanumerical text variables (boolean or word) select the icon in the toolbar and position the text box on the screen. This box is used to display the value. The position and size can be changed later.	 
2	Select the **Variable** to be displayed (use the **…** button to browse for a variable). Define the attributes of the variable on the dialog tabs.	

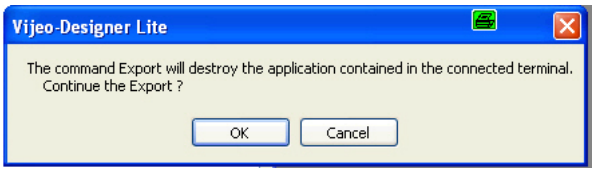
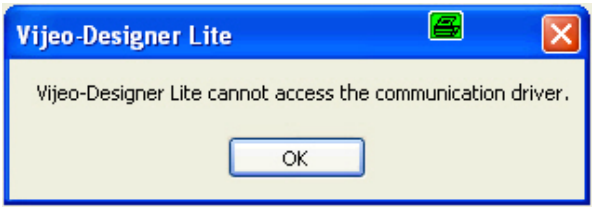
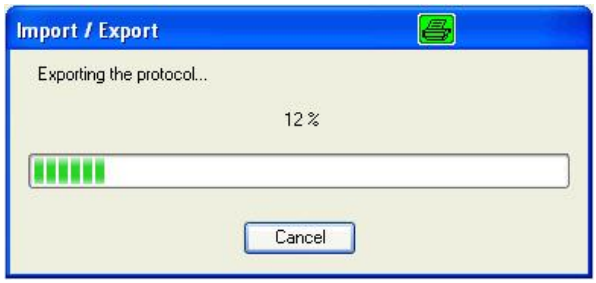
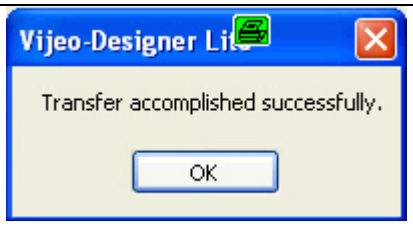
3	Define the type of variable (%Mx, %MWx, etc.), the number (**i:**), and the **Format** (bit, word, etc.). If you have imported variables from other devices (using Equipment -> Symbol Variables File), it is possible to select a variable from the **Symbol** list for that device. Confirm with **OK**.	 
4	Use **Display Type** to define how the variable is to be displayed. In the **Conversion,** it is possible to enter a fomula tor adjust the value before it is displayed.	
5	In the **Access Mode and Security** tab, the access and the security level can be assigned.	
6	For an enumerated list, select the icon in the toolbar and position the text box on the screen. This box is used to display the value. The position and size can be changed later.	

7	On the **Enumerated List – Settings** tab, select the variable. Input the **Variable** address and **Font Size**.	
8	Use the tab **List Value** to assign the different values of the variable to different texts. The texts are displayed at run time on the screen according to the variable's actual value. If you selected more than one language you have to enter texts for each language.	 

Screen Navigation & Viewing System Pages

1	Select the highlighted icon as shown in the image and whilst pressing the left mouse-key, draw a rectangle. This rectangle is to be defined as an active touch-field. If an operator touches this field, the action defined for it is invoked. In the **Button-Settings** set the button up as follows: For the **Button Action,** select **Access Application Panel** in the drop down list. On **Panel Access Functions Settings** choose the corresponding screen number to switch to when the button has been touched. An image for the button can be chosen after pressing the highlighted button **(…)**.	  

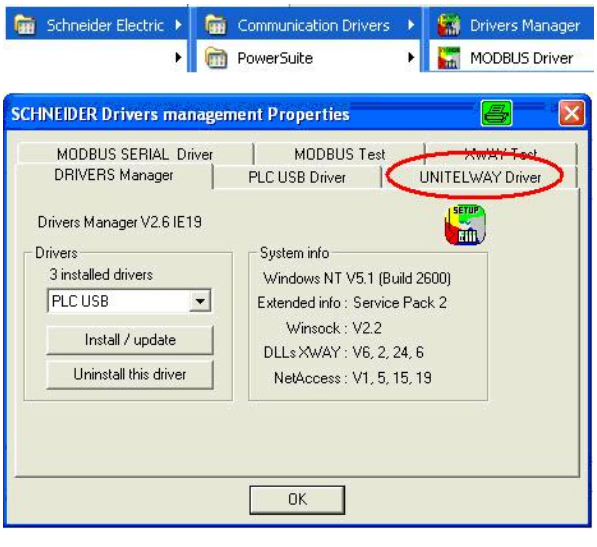
Open an existing project and transfer to the HMI	**1**	Select: **File->Open**. In the following path: *C:\Program-Files\Schneider Electric\VIJEO-DESIGNER LITE* Select the file: **Optimized_CANopen_Twido.dop** and open it. The application is loaded and the structure and screens are shown. Check the connection between the HMI and your PC.	  
	2	Downloading the project to the HMI is initiated with: **Device**->**Download**	
	3	In the next window the possible language files are shown. Select the desired language by clicking on it. Click on **Download…**	

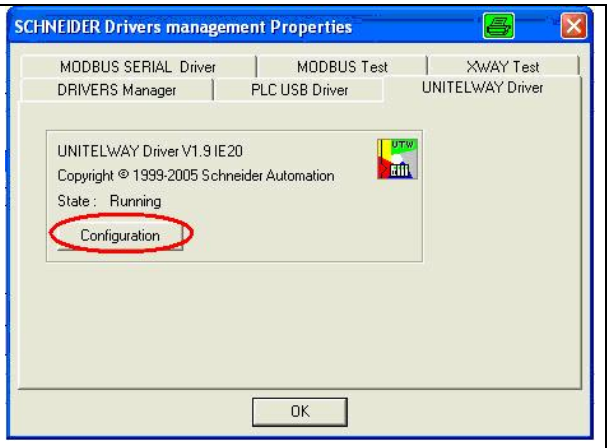
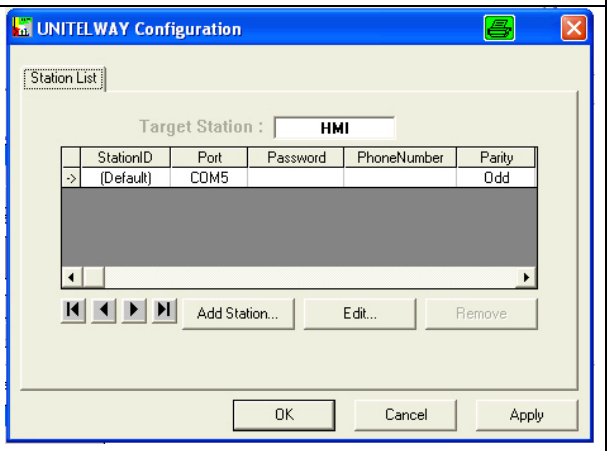
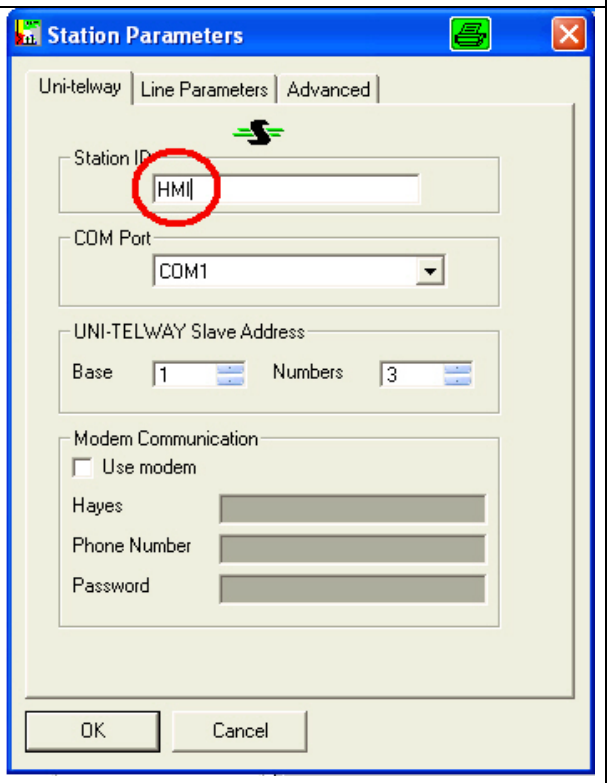
4	A pop-up appears with a message. Continue by clicking on **OK.** **Note:** If the communications driver is not installed, change the parameter in the Windows Driver-Manager (see the description at the end of this chapter).	 
5	The download starts.	
6	The progress is indicated with a progress bar.	
7	The transfer is finished. Continue by clicking **OK.**	

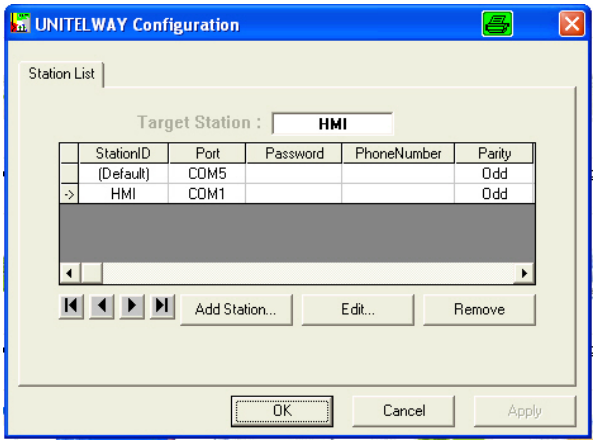
Drivers-Manager

In case of a communication driver problem (as in the previous section), follow these steps to modify the driver using the drivers manager.

Add an interface

1	The **Drivers Manager** is a separate tool used to adjust the set up of the communication drivers. Start the tool under Windows with: **Start->All Programs-> Schneider Electric -> Communication Drivers -> Drivers Manager** and go to the **UNITELWAY Driver** tab.	

2	Click on **Configuration.**	
3	A list of available interfaces is displayed, for example COM5. For the serial interface with the XBT-Z915 cable, you have to add COM1. Click on **Add Station…**	
4	Select COM1 as **COM Port.** For the **Station ID** input **HMI**. Close the dialog with **OK.**	

5	The second interface is now visible. Close the dialog with **OK.**	
6	The drivers are reset and initialized.	
7	The reset is confirmed. Continue by clicking **OK**.	

Devices

Introduction

This chapter describes the steps required to initialize and configure the devices to attain the described system function.

Altivar 312

General

The parameters for the ATV312 can be set using the SoMove Lite configuration software. This section describes the SoMove Lite procedure.

Before carrying out the steps described below, you must verify that:

- The SoMove Lite configuration software is installed on your PC.
- The variable speed drive is connected to the power supply.
- The PC is connected to the variable speed drive via the communication cable.

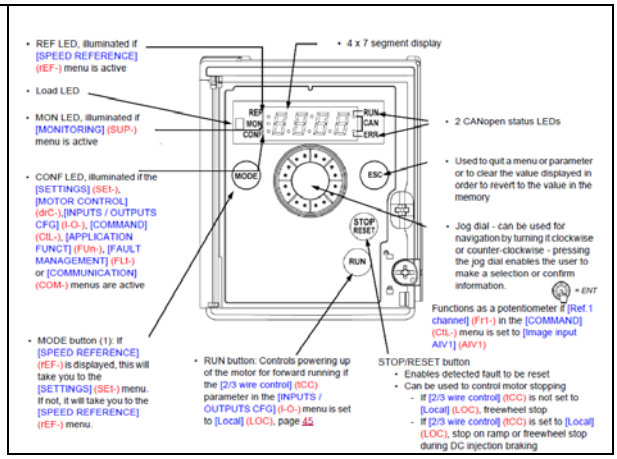
The following steps are described:

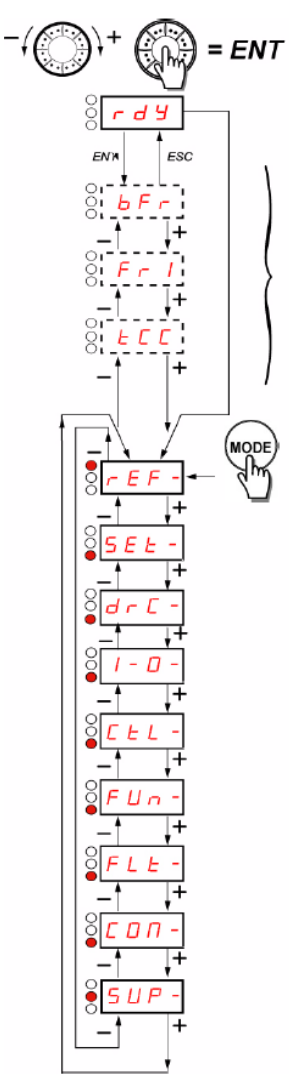
- ATV312 Control Panel
- Configuring ATV312 with SoMove Lite
- Download the configuration
- Connect to device/ Upload the configuration

ATV312 Control Panel

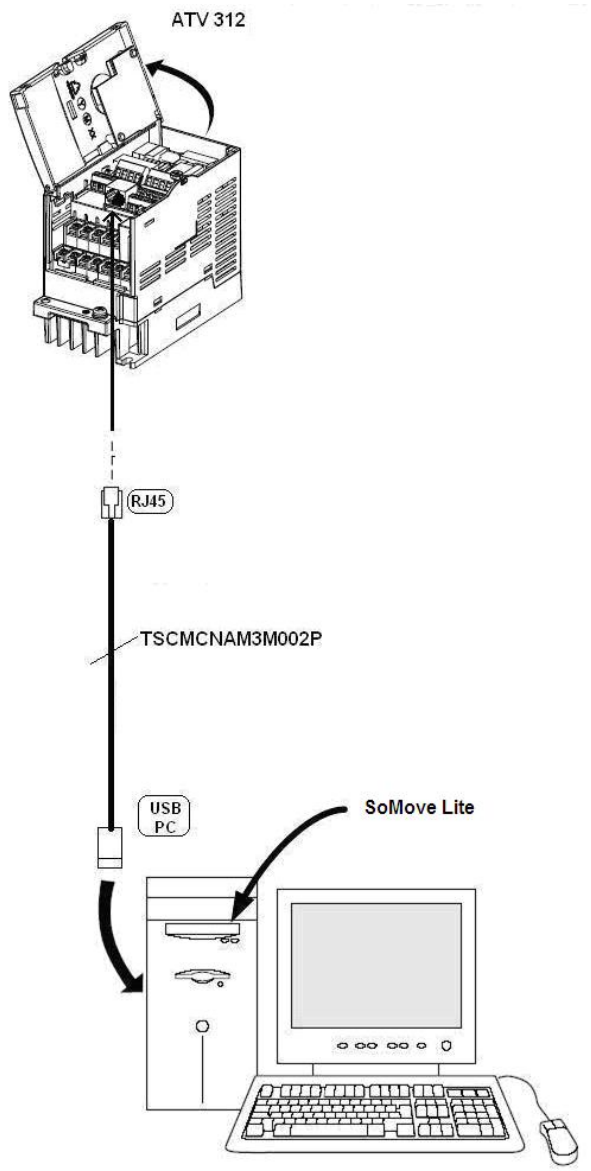
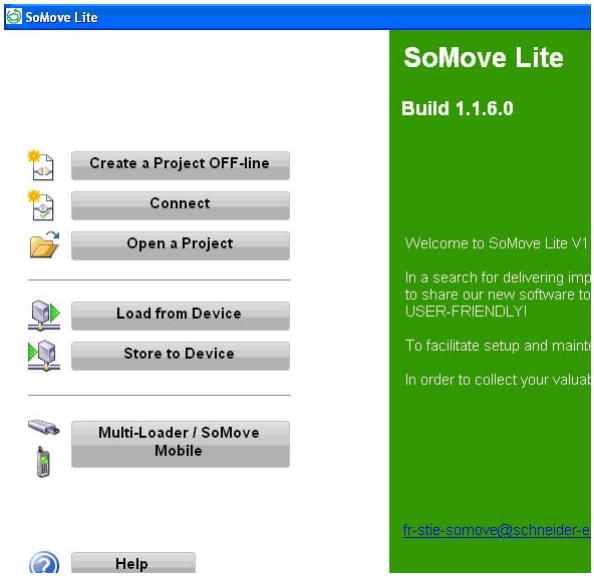
The ATV312 parameters can be entered or modified via the control panel on the front of the device. This section describes how to set up the drive using this control panel.

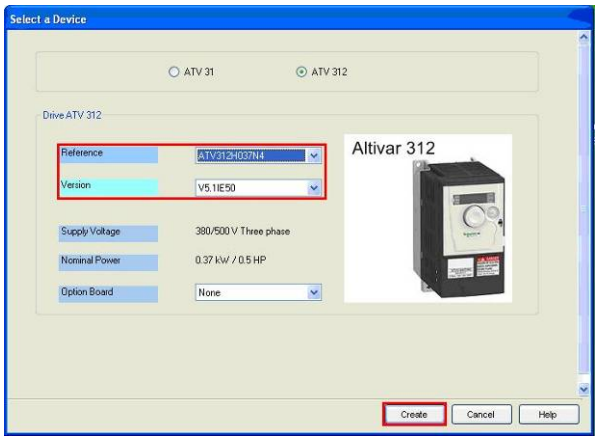
Alternatively, you can use the SoMove Lite software to configure the CANopen addresses and baud rates.

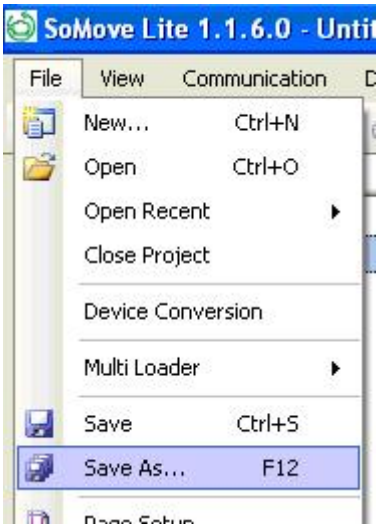
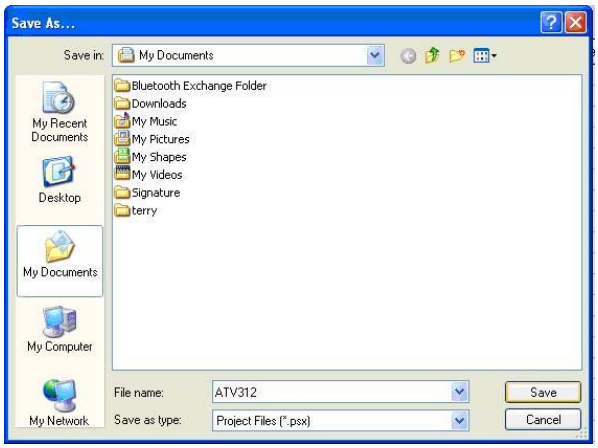
1	The CANopen address and Baudrate can be input using the buttons and the jog dial on the front panel of the Altivar.	

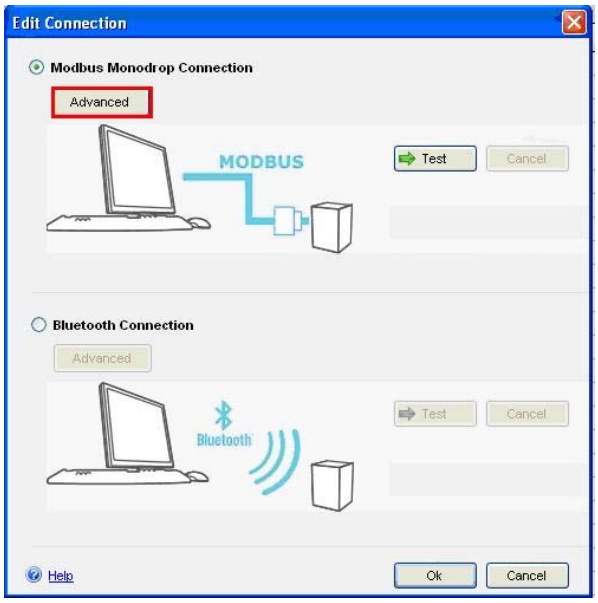
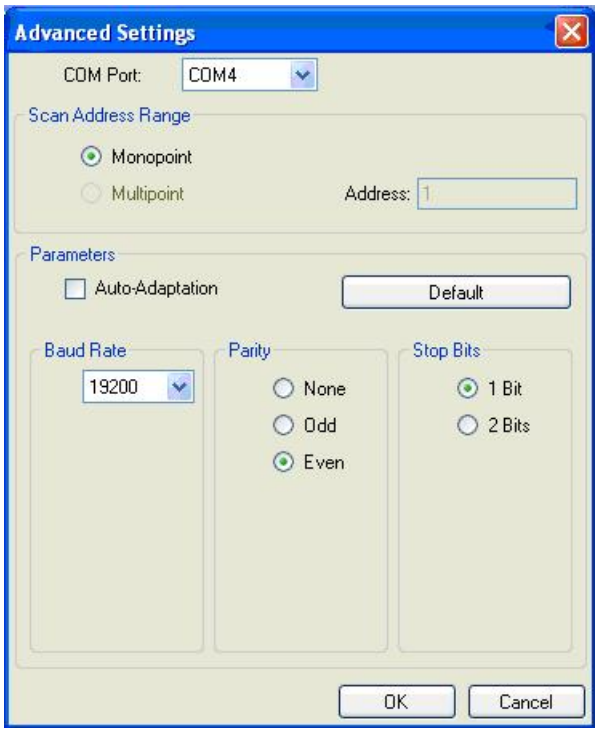
2	Using the buttons on the front panel, select the sub-menu **Communication**.	 These three parameters are only visible when the drive is powered up for the first time. The settings can be amended subsequently in the menus: [MOTOR CONTROL] (drC-) for [Standard mot. freq] (bFr), [COMMAND] (CtL-) for [Ref.1 channel] (Fr1) [INPUTS / OUTPUTS CFG] (I-O-) for [2/3 wire control] (tCC) [SPEED REFERENCE] (rEF-) [SETTINGS] (SEt-) [MOTOR CONTROL] (drC-) [INPUTS / OUTPUTS CFG] (I-O-) [COMMAND] (CtL-) [APPLICATION FUNCT.] (FUn-) [FAULT MANAGEMENT] (FLt) [COMMUNICATION] (COM-) [MONITORING] (SUP-)
3	In the **Communication** sub-menu input the CANopen address in the parameter **AdC0.** In the example application the adresses for the four controllers are **1** to **4.**	
4	Also in the **Communication** sub-menu, in the parameter **BdC0,** set the baudrate to **500.0** (kBits).	
6	**For the ATV312 to operate with the new address or Baudrate, a power cycle (on, off, on) is required.**	

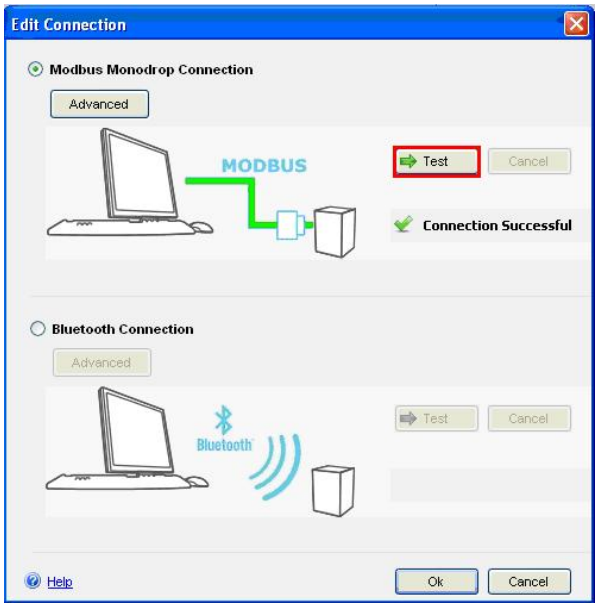
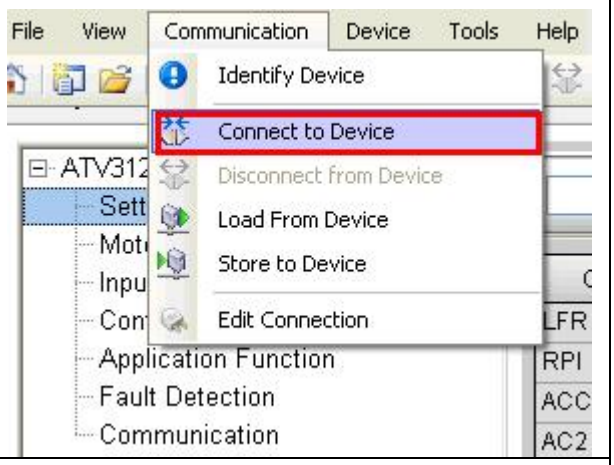
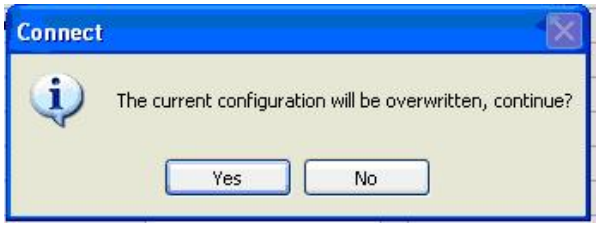
Configuring ATV312 with SoMove Lite

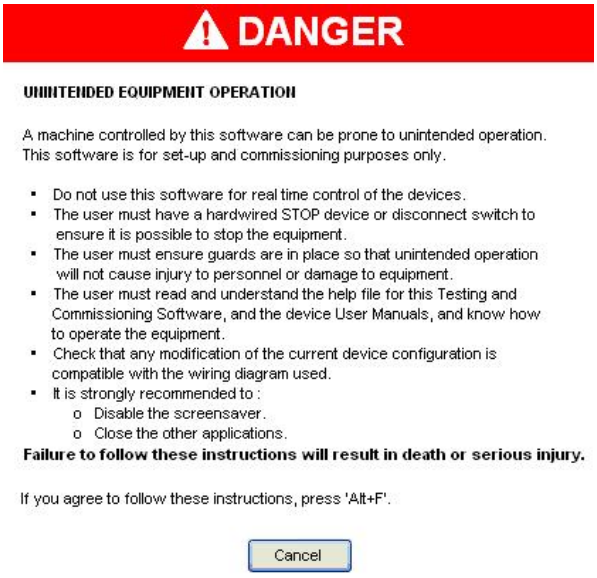
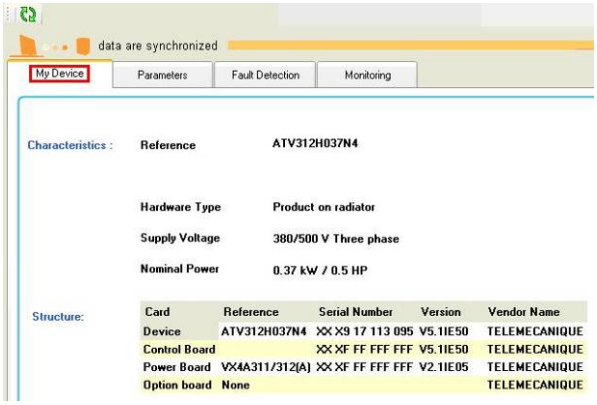
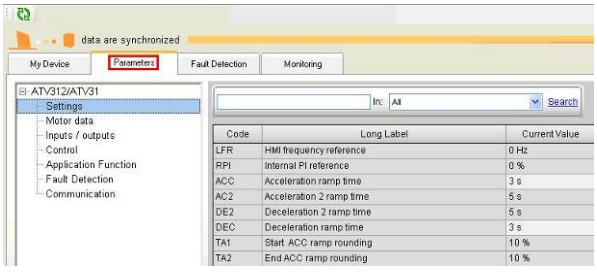
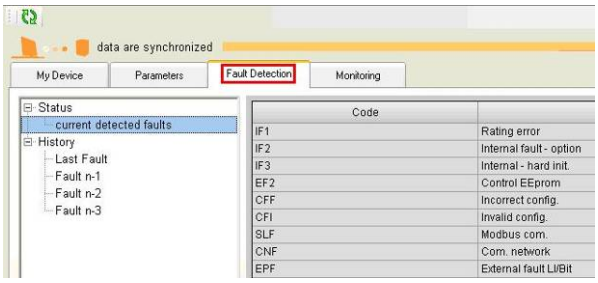
1	First connect the PC to the drive. Use the programming cable, **TSCMCNAM3M002P** to connect the USB port on your PC to the RJ45 socket on the Drive. If you used CANopen taps, verify that the PC is connected to the middle socket on the tap and that only the target drive is connected to the tap.	
2	After starting SoMove Lite you will see the window as shown here on the right – the version is automatically shown in English.	

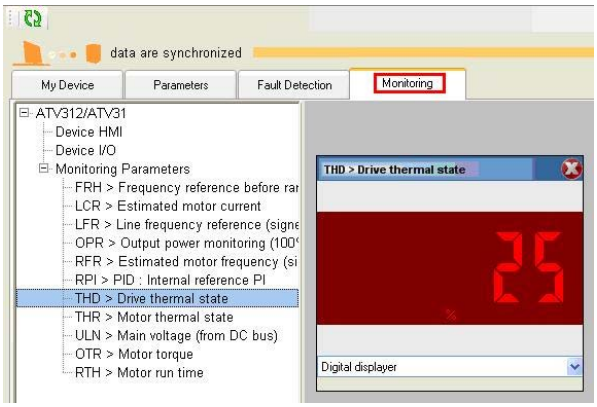
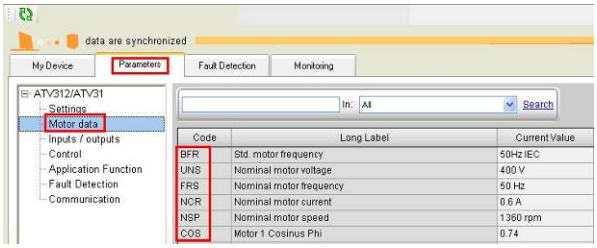
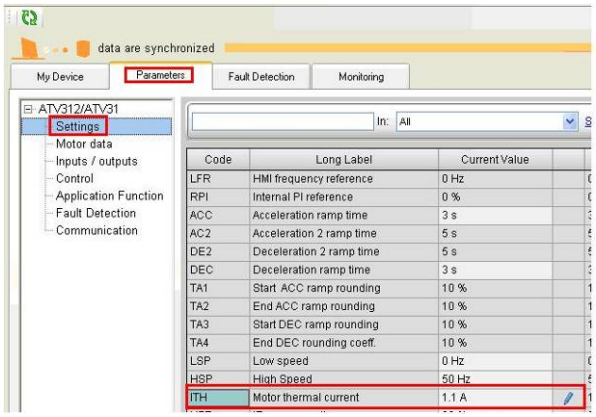
3	To create a new Project, select **Create a Project OFF-line** and choose the desired device. Continue with **Next.**	
4	Select **ATV312** from the options. In the **Device** dialog, select a **Reference-Type** and the **Firmware Version.** In this example: **Reference:** ATV12H037N4 **Version:** V5.1IE50 Additional information displayed: - Supply Voltage - Nominal Power of the drive. Continue with **Create.**	

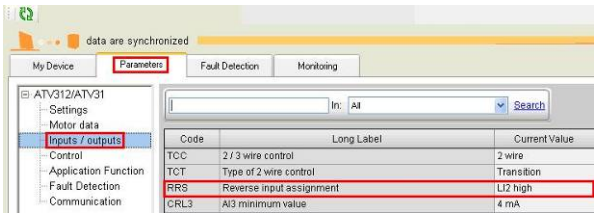
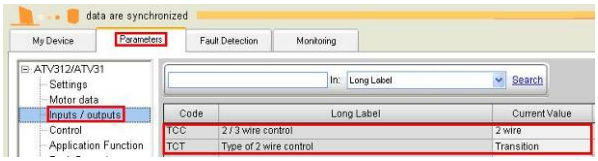
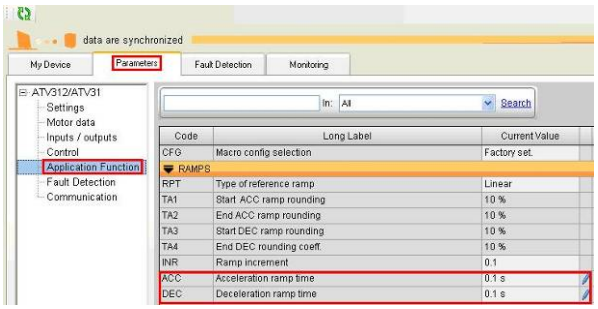
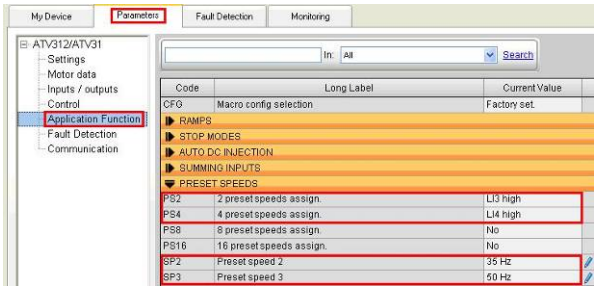
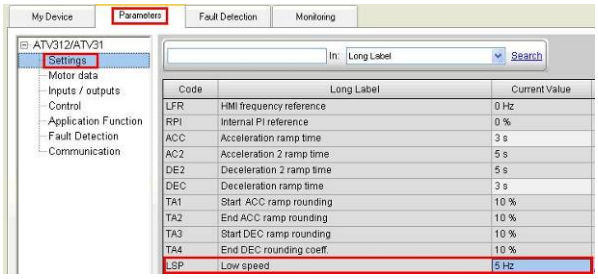
5	To save the project use **File–>Save as…** Insert the **File name** (here ATV312) and continue with **Save**	 
6	Select: **Communication->Edit Connection** to go to the connection dialog.	

7	Select **Advanced** to check and edit the connection settings.	
8	Select the appropriate **COM Port**, here **COM4** is used. The port can be changed as per the requirement. Here default settings have been used. **Baud Rate:** 19200 **Parity:** Even **Stop Bits:** 1 Continue with **OK**.	

9	You can also perform a test of the connection by clicking on **Test** Then select **OK.**	
10	Select: **Communication->Connect to Device** to go to the connection dialog.	
11	If the configuration of the device is not same as given in the file. Then user gets the following warning. Click **Yes.**	

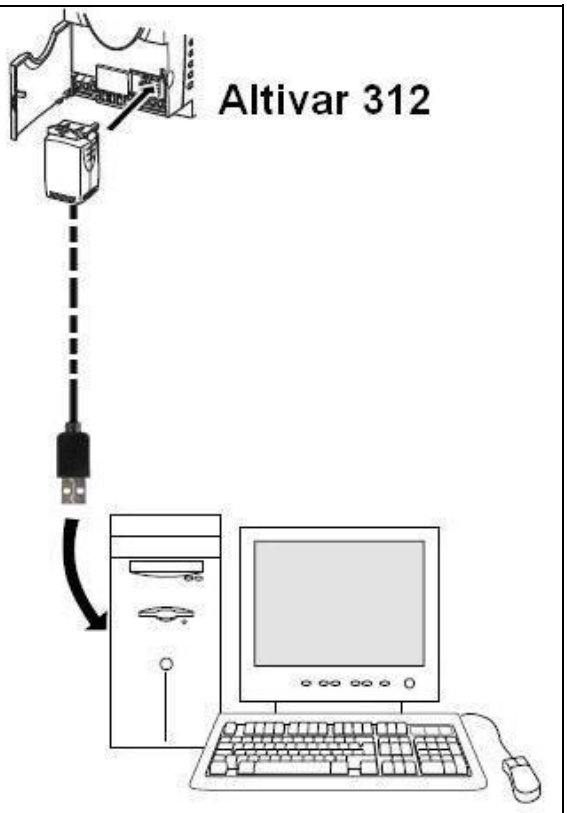
12	Press **Alt + F** to continue.	
13	The Main-Window of SoMove consists of four tabs. The tab, **MyDevice**, includes the basic information about the device.	
14	The tab, **Parameters**, shows a table of all adjustable parameters.	
15	The tab, **Fault Detection,** shows the **current detected faults**.	

16	The tab **Monitoring,** is used to monitor the parameter values of the drive. Click on the parameter to be observed and then click again on the plane on the right hand side.	
17	On the **Parameters** tab, select **Motor Data**. Leave the value for **BFR** at 50Hz. Insert your motor values as found on the name plate of the motor.	
18	Select**:** **Settings** ➔ **Motor Thermal Current** Modify **ITH** according to the values for your motor.	

19	Select: **Inputs/Outputs** ➔ Set the input for **RRS** to **LI2 high.** ➔ **Set TCC to 2 wire** ➔ Set **TCT** to **Transition** Select: **Application Function** ➔ **RAMPS** Change **ACC** & **DEC** to 0.1s ➔ **PRESET SPEEDS** Set the Input for **PS2 (2 preset speeds assign.) -> LI3 high** and **LI4 high**. The Preset speeds can be chosen by setting the appropriate inputs of the drive. Change **SP2 (Preset Speed 2)** to 35Hz and SP3 **(Preset Speed 2)** to 50Hz. Select: **Settings** ➔ Set **LSP (Low Speed)** to **5 Hz** .	    

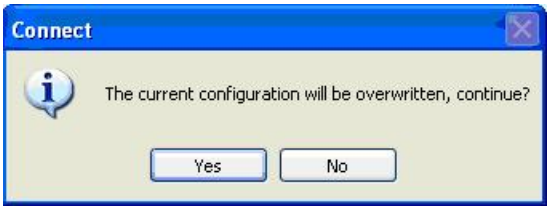
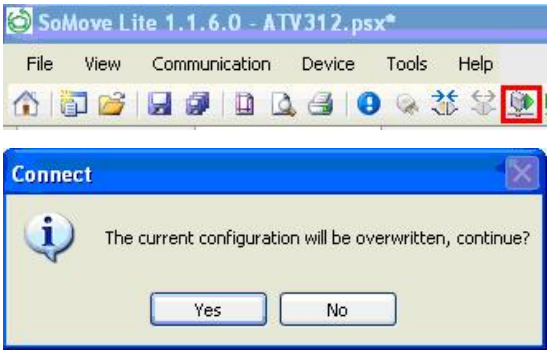
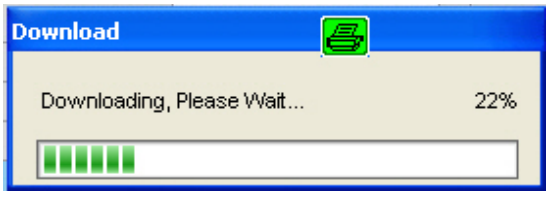
20	Select: **File**-> **Save** or use the icon in the toolbar, to save your configuration.	

Download the Configuration

1	Check that the PC is connected to the Altivar drive. For this use the cable set **TSCMCNAM3M002P**.	
2	To download the configuration to the drive click on icon in the toolbar: **Store Values to Device**	

3	A progress bar is displayed during the download. Click OK.	

Connect to Device / Upload the Configuration

1	To connect to the device click the *Connect to Device* icon in the toolbar Or select: **Communication->Connect to Device**	
2	To upload the configuration to the PC answer the question that follows with **YES** The configuration in the PC-File will be overwritten!	
3	You can also upload the configuration from the device into PC by clicking on the *Load Values from Device* icon And answer the question with **YES**	
4	A progress bar is displayed during the upload.	

Appendix

Detailed Component List

The following tables are a list of the main components of the Optimized CANopen Twido architecture. The complete bill of materials is included in the EPLAN file for Optimized CANopen Twido.

	Hardware-Components				
	Pos.	**Qty.**	**Description**	**Part Number**	**Rev./ Vers.**
Sarel cabinet	1.1	1	Cabinet, 1200 x 800 x 400 mm (Sarel)	83362	
	1.2	1	Fan filter, 230 Vac, 56 m³/h (Sarel)	87901	
	1.3	2	Filter unit for ventilation (Sarel)	87911	
	1.4	1	Thermostat for Fan, 0..60°C (Sarel)	87562	
Main switch	2.0	1	Compact Master switch 3 pole 400 Vac	NSX100F	
Power supply	3.0	1	Phaseo Power supply 230 Vac / 24 Vdc 5 A	ABL8RPS24050	
Controller and I/O modules	4.0	1	Twido modular with 12 x In / 8 x Out	TWDLMDA20DRT	FW 4.20
	4.1	1	CANopen master interface	TWDNCO1M	
	4.2	1	TM2 digital 8 out /16 input	TM2DMM24DRT	
	4.3	1	TM2 analog 2 out / 4 input	TM2AMM6HT	
	4.4	1	Analog extension card	TM2ALM3LT	
Drives & Motor Starter	5.0	4	ATV312 variable speed drives 0.37 kW	ATV312H037N4	
	5.1	1	TeSysU Base unit for two directions	LU2B12BL	
	5.2	1	TeSysU Control unit	LUCA05BL	
	5.3	1	TeSysU CANopen interface	LULC08	
Sensor	6.0	2	Preventa guard switch	XCS-A502	
HMI	7.0	1	HMI Magelis semi-graphical touch display	XBTRT511	
E-Stop	8.0	1	Preventa Emergency Stop safety relays	XPSAC5121	
	8.1	1	Harmony Emergency Stop pushbutton	XB5AS844	
	8.2	4	TeSys Motor circuit breaker 2.5 A	GV2L07	
Pushbutton	9.0	1	Harmony Box for 1 button	XALD01	
	9.1	2	Harmony Signal lamp LED white	XB5AVB1	
	9.2	1	Harmony Pushbutton with LED red	XB5AW34B5	
	9.3	2	Harmony Pushbutton with LED blue	XB5AW36B5	

Software and cable

Software-Components				
Pos.	**Qty.**	**Description**	**Part Number**	**Rev./ Vers.**
10.0	1	TwidoSuite Software	TWDBTFU10M	V2.10
10.1	1	USB programming cable	TSXCUSB485	
10.2	1	Communication cable Twido-HMI	XBTZ9680	
10.3	1	Vijeo Designer Lite	VJDSNDTMSV12M	V1.2
10.4	1	Programming cable PC-HMI	XBTZ915	
10.5	1	SoMove Lite	VW3A8200	V1.1.6
10.6	1	Altivar-Set Connection cable for drives	TSCMCNAM3M002P	

Component Protection Classes

Positioning Protection class	Component	In Field, On site			Cabinet		
					Front		inside
		IP54	IP65	IP67	IP55	IP65	IP20
	Harmony Emergency Stop switch				X		X
	Emergency Stop switch housing		X				
	Master Switch NSX 100F						X
	Rotating Door handle for master switch				X		X
	TeSys motor protection devices and load contactors						X
	Contactors and fuses						X
	Phaseo Power supply						X
	Emergency stop switches XPSAC***						X
	Twido controller and expansion modules						X
	Magelis XBTRT Touch panel					X	X
	Altivar 312 variable speed drive						X
	CANopen Taps						X
	Switch series ”Harmony“ in housing		IP66				
	Filter and Filter fan				IP40		
	Fan-Thermostat						X

Component Features

Components

Controller Twido

This example uses a Twido modular Controller with TwidoSuite software and programming cable set (TWDLMDA20DRT+ TWDBTFU10EF + TSXPCX1031):

- 24 Vdc
- 12 digital inputs
- 8 digital relay outputs
- expandable up to 7 modules

2 Programming Languages:

- Ladder (LD)
- Instruction List (IL)

Pre-defined functions:

- Drum control
- Fast counters up to 5kHz
- Very fast counters up to 20 kHz
- Frequency measuring 1..20 kHz
- Reserved Memory for LIFO/FIFO-processing
- PWM-/PLS-Outputs
- External Controller-Start
- PID-controller

CANopen master module TWDNCO1M

Master module for Twido Controllers with:

- Control of up to 16 Slaves (Depends on number of PDOs/SDOs)
- Baudrates 125 / 250 / 500 kBit/s
- Slave watchdog with Node Guarding or Heartbeat
- Configuration tool integrated in Twidosuite
- Bus configurable through the backplane of the Controller
- Integrated macros for quick installation
- Slim format (30mm width)
- Pluggable contacts for power supply
- Configuration by Controller during power up

Components cont.

Twido analog module TM2AMM3HT

4 x analog input 0..10 Vdc

4096 increments, 12-bit digital resolution

2x analog output 0..10 Vdc

4096 increments, 12-bit digital resolution

Safety Module: Preventa XPSAC5121

Main technical characteristics:

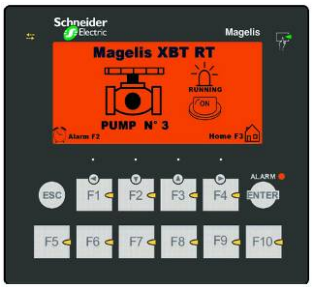
For monitoring	Emergency stop
Max. category accord. EN954-1	4
No. of safety circuits	3 N/O
No. of additional circuits	-
Indicators	3 LED
Power supply AC/DC	24 V
Synchro time between inputs	Infinite
Response time	< 40 ms
AC-15 breaking capacity	C300
DC-13 breaking capacity	24 Vdc /1.5 A - L/R 50 ms
Minimum voltage and current	17 V / 10 mA
Dimensions (mm)	114 x 22.5 x 99
Connection	Captive screw-clamp terminals
Degree of protection	IP20 (terminals) IP40 (casing)

Components
Contd.

Power Supply Phaseo: ABL8RPS24050

- 100..240 Vac / 24 Vdc
- 5 A
- Slim design
- Parallel wiring possible
- Short circuit and power surge protected

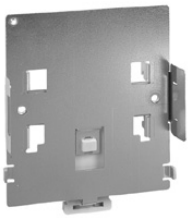
Magelis XBTRT511 HMI touch terminal



- Touch screen (TFT-Technology) with 24 Vdc power supply
- 13 signal LEDs
- 10 function keys
- Brightness and Contrast adjustment
- Communication via Uni-Telway, Modbus master & slave
- Flat Profile
- Temperature range: 0..+ 50°C
- Certificates: UL, CSA

Altivar Variable speed drive: ATV312H037N4

- 0.37 kW 380..500 Vac 3 phase
- Integrated EMC Filter
- Temperature range: - 10..+ 50°C
- Speed Range: 1 to 20 (0...200 Hz)
- Speed control using Flow Vector control
- Modbus and CANopen compatible
- 2 analog inputs, 1 analog output
- 6 Digital inputs
- 2 - 3 digital status outputs possible
- Drive and motor protection
- Compact format, side by side installation possible, using adapter VW3A11852 can be mounted on a DIN rail



TeSysU Motor starter

with CANopen interface

TesysU LU2B12BL+ LUCA05BL + LULC08

Magnetic Circuit breaker: GV2L07 and GV2L08

- 2.5 A / 4.0 A
- Short circuit protected
- Magnetic cut off at 33.5 A
- lockable

TeSysD Contactor LC1D09BD

- Rated current 9 A AC3
- 1x NO contact
- 1x NC contact
- Positive opening operation allows for use in functional safety circuits
- 24 Vdc control voltage incl. suppressor circuit

Contact

Publisher	Process & Machine Business OEM Application & Customer Satisfaction Schneider Electric Automation GmbH Steinheimer Strasse 117 D - 63500 Seligenstadt Germany
Homepage	http://www.schneider-electric.com/sites/corporate/en/home.page
As standards, specifications and designs change from time to time, please ask for confirmation of the information given in this publication.	